
OpenVMS Alpha Version 7.3–1 New Features and Documentation Overview

Order Number: AA–RSHYA–TE

June 2002

This manual describes the new features associated with the OpenVMS Alpha Version 7.3–1 operating system and provides an overview of the documentation that supports the software.

Revision/Update Information: This is a new manual.

Software Version: OpenVMS Alpha Version 7.3–1

**Compaq Computer Corporation
Houston, Texas**

© 2002 Compaq Information Technologies Group, L.P.

Compaq, the Compaq logo, AlphaServer, OpenVMS, POLYCENTER, Tru64, VAX, VMS, and the DIGITAL logo are trademarks of Compaq Information Technologies Group, L.P. in the U.S. and/or other countries.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the U.S. and/or other countries.

Motif, UNIX and X/Open are trademarks of The Open Group in the U.S. and/or other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

ZK6657

The Compaq *OpenVMS* documentation set is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1b.

Contents

Preface	ix
Part I OpenVMS Version 7.3–1 New Features	
1 Summary of OpenVMS Alpha Version 7.3–1 New Features	
1.1 Performance Features	1–1
1.2 Security Features	1–3
1.3 UNIX Portability Features	1–3
1.4 System Management Tools	1–3
1.5 Summary Table	1–4
2 Compaq OpenVMS e-Business Features	
2.1 Attunity Connect "On Platform" Package	2–2
2.2 Compaq BridgeWorks	2–2
2.3 COM Version 1.2 for OpenVMS	2–2
2.4 Compaq Enterprise Directory for e-Business	2–3
2.5 NetBeans for OpenVMS	2–3
2.6 Reliable Transaction Router (RTR)	2–3
2.7 Compaq Secure Web Browser for OpenVMS Alpha (based on Mozilla) ...	2–3
2.8 Compaq Secure Web Server for OpenVMS Alpha (based on Apache)	2–4
2.9 Simple Object Access Protocol (SOAP) Toolkit	2–4
2.10 Compaq Software Development Kit (SDK) <i>for the Java™ Platform</i>	2–5
2.11 Compaq XML Technology	2–5
3 General User Features	
3.1 Documentation Updates on the World Wide Web	3–1
3.2 New and Archived Manuals	3–1
3.3 3Dlabs Oxygen VX1-AGP4X 2D Graphics Support	3–2
3.4 DCL Commands and Lexical Functions	3–2
3.4.1 RECALL/ALL Update	3–4
3.4.2 SET COMMAND/OBJECT Update	3–4
3.4.3 SHOW DEVICES Update	3–5
3.5 WWPPS Adds Support for GB18030 Codeset	3–5

4 System Management Features

4.1	ACCOUNTING Utility Has New Qualifier	4-1
4.2	AlphaServer DS25	4-1
4.2.1	Broadcom 5700 Series Gigabit Ethernet	4-1
4.2.2	Use New CDRECORD.COM to Create a CD-ROM	4-1
4.3	ANALYZE/DISK_STRUCTURE Utility Has New Qualifier	4-2
4.4	Authentication and Credentials Management Extensions (ACME) Subsystem	4-2
4.5	BACKUP Utility Supports New Keywords for /DENSITY Qualifier	4-3
4.6	Batch Job Support for NUMA Resource Affinity Domains	4-3
4.6.1	Batch Queue Level RAD Support	4-3
4.6.2	Job-Level RAD Support	4-4
4.6.3	Run-Time Behavior	4-5
4.6.3.1	Error Processing	4-5
4.6.3.2	RAD Modifications on Batch Queues	4-6
4.7	DECram Support for Galaxy	4-6
4.8	DECram, Mailbox Driver, and Volume Shadowing Performance Gains	4-6
4.9	Fast Path Performance Enhancements	4-7
4.9.1	Distributed Interrupts	4-7
4.9.2	Fast Path Code and RAD	4-8
4.10	File Service Extensions	4-8
4.10.1	Additional File Times	4-8
4.10.1.1	ACP/QIO Access Dates	4-9
4.10.1.2	Compaq C Run-Time Library Dates	4-9
4.10.2	DCL Access Dates	4-9
4.10.3	Hard Links	4-10
4.10.4	Case-Sensitive File Operations	4-12
4.10.4.1	Record Management System (RMS) Case Sensitivity	4-13
4.10.4.2	ACP/QIO Case Sensitivity	4-13
4.10.4.3	System Services That Control the Process Case Sensitivity	4-13
4.10.4.4	DCL Case Sensitivity	4-14
4.10.4.5	C RTL Support	4-14
4.11	New Compaq Graphical Configuration Manager (GCM)	4-16
4.12	Kerberos for OpenVMS	4-16
4.13	New LANCP Qualifiers for SET DEVICE and SHOW DEVICES	4-17
4.14	LIBDECOMP.COM Enhancements	4-17
4.15	ODS-5 Volume Structure	4-20
4.16	OpenVMS Cluster New Features	4-20
4.16.1	OpenVMS Cluster Performance Improvements	4-20
4.16.2	Fibre Channel Driver Optimizations	4-22
4.16.3	I/O Interrupt Coalescing for Fibre Channel Configurations	4-23
4.16.4	MSCP Served Devices: Increase from 512 to 1000	4-23
4.16.5	Multipath Failover to an MSCP-Served Path in SCSI and Fibre Channel Configurations	4-23
4.16.6	Automatic Multipath Balancing for Fibre Channel and SCSI Devices	4-24
4.16.7	Multipath Tape Support, Including Failover, for Fibre Channel Configurations	4-24
4.16.8	New Cluster SCA Circuit and Port Functionality	4-25
4.16.8.1	Ability to Set Port and Circuit Priorities	4-25
4.16.8.2	Ability to Enable/Disable PEdriver Checksumming	4-25
4.16.8.3	New SCS Dynamic Load Class Support	4-26
4.17	OpenVMS Registry New Features	4-26
4.17.1	Database Version Support	4-26

4.17.2	Value Type Support	4-26
4.17.3	New Registry Database Version	4-27
4.18	SHOW CLUSTER Utility Adds New Fields	4-27
4.19	New Method for Naming Spawned Subprocesses	4-28
4.20	SYSMAN Utility Has New Commands and Qualifiers	4-28
4.21	New System Parameters	4-29
4.21.1	DCL_CTLFLAGS	4-29
4.21.2	DELPRC_EXIT	4-29
4.21.3	MPDEV_AFB_INTVL	4-30
4.21.4	RMS_SEQFILE_WBH	4-30
4.22	New Capabilities for SCACP Utility	4-30
4.22.1	Cluster Ports and Circuits	4-30
4.22.2	New Commands and Qualifiers	4-30
4.23	Ultra3 SCSI Adapter Supported with Fast Path	4-32
4.24	XA Capability Added to DECdtm	4-32

5 Programming Features

5.1	Taking Advantage of Performance-Enhancing Features of Alpha Processors	5-1
5.2	ANALYZE Usage Summary Has New Qualifier	5-1
5.3	Compaq C Run-Time Library Enhancements	5-1
5.3.1	Files Larger Than 2 Gigabytes	5-1
5.3.2	Enabling C RTL Features at Compile/Link Time	5-2
5.3.3	Enhanced Support for UNIX Style File Names	5-2
5.3.4	POSIX Root Support	5-3
5.3.5	32-Bit Group Identifiers	5-4
5.3.6	New and Changed C RTL Functions	5-5
5.4	DECdtm System Services	5-5
5.5	Open Source Security for OpenVMS—CDSA and SSL	5-6
5.6	Using LDAP with Compaq SSL for OpenVMS	5-7
5.7	New Alpha Floating-Point Registers and Execution Data Block (FRED)	5-8
5.8	Future Architecture-Specific Code Changes	5-8
5.9	Support for Image Sections in Galaxy Shared Memory	5-10
5.10	Larger Mailbox Buffer Quotas	5-10
5.11	RMS New Features	5-11
5.11.1	Increase in Default I/O Transfer Size to Reduce I/O	5-11
5.11.2	Support for POSIX-Compliant File Timestamps	5-11
5.11.3	New System RMS Write-Behind Performance Option	5-12
5.12	SDA New Features	5-12
5.12.1	New VALIDATE TQE Command	5-12
5.12.2	New TQE Type	5-12
5.12.3	New Qualifiers	5-13
5.12.4	New SDA Extended File Cache (XFC) Commands	5-13
5.13	New Qualifiers for SDA Spinlock Tracing Utility	5-16
5.14	New and Updated System Services	5-16
5.15	System Services Updates for Batch Job Support for NUMA RADs	5-18
5.15.1	\$GETQUI Updates	5-18
5.15.1.1	New \$GETQUI Item Code	5-19
5.15.1.2	New RAD Flag for QUI\$_QUEUE_FLAGS and QUI\$_JOB_FLAGS Item Codes	5-19
5.15.2	\$SNDJBC Updates	5-19
5.16	NUMA Support Added to Threads Libraries	5-19

6 Associated Products Features

6.1	Compaq Advanced Server for OpenVMS Alpha	6-1
6.2	Compaq C Run-Time Header Files Library	6-2
6.3	Compaq PATHWORKS for OpenVMS (Advanced Server)	6-2
6.4	Compaq TCP/IP Services for OpenVMS Version 5.3 New Features	6-3
6.5	OpenVMS Management Station Version 3.2	6-4
6.6	Powerstorm 300/350 Graphics Support Version 2.0 for OpenVMS	6-4
6.7	UNIX Portability Features	6-4
6.8	Open Source Tools for OpenVMS CD-ROM	6-4

Part II OpenVMS Documentation

7 OpenVMS Documentation Overview

8 OpenVMS Printed and Online Documentation

8.1	Printed Documentation	8-1
8.1.1	OpenVMS Media Kit Documentation	8-1
8.1.2	OpenVMS Documentation Sets	8-2
8.1.3	Documentation for System Integrated Products	8-4
8.1.4	Archived OpenVMS Documentation	8-4
8.2	Documentation Pilot Project	8-5
8.3	Online Documentation on CD-ROM	8-5
8.3.1	Online Formats	8-5
8.3.2	PDF Reader	8-6
8.4	Online Documentation on the OpenVMS Web Site	8-6
8.5	Online Help	8-6

9 Descriptions of OpenVMS Manuals

9.1	Manuals in the OpenVMS Media Kit	9-1
9.2	Manuals in the OpenVMS Base Documentation Set	9-2
9.3	Additional Manuals in the OpenVMS Full Documentation Set	9-3
9.4	OpenVMS Alpha Device Driver Manual	9-7
9.5	RMS Journaling Manual	9-7
9.6	Archived Manuals	9-7

Index

Tables

1-1	Summary of OpenVMS Alpha Version 7.3-1 Software Features	1-4
4-1	SET DEVICE Command Qualifiers	4-17
4-2	SHOW DEVICES Command Qualifiers	4-17
4-3	Libraries Recognized by LIBDECOMP.COM	4-18
4-4	CIRCUITS Class New Fields	4-27
4-5	LOCAL_PORTS Class New Fields	4-27
4-6	New SYSMAN Command Qualifiers	4-29
4-7	New SYSMAN Utility Commands	4-29
4-8	New SCACP SET and SHOW Commands and Their Qualifiers	4-31

4-9	New SCACP Command Qualifiers	4-32
5-1	DECdtm System Services	5-5
5-2	New SDA Command Qualifiers	5-13
5-3	New SDA Extended File Cache (XFC) Commands	5-13
7-1	Documentation Set Changes for OpenVMS Version 7.3-1	7-1
8-1	OpenVMS Media Kit Manuals	8-2
8-2	OpenVMS Full Documentation Set (QA-001AA-GZ.7.3-1)	8-2
8-3	System Integrated Products Documentation	8-4
9-1	Archived OpenVMS Manuals	9-8
9-2	Archived Networking Manuals and Installation Supplements	9-9

Intended Audience

This manual is intended for general users, system managers, and programmers who use the Compaq *OpenVMS* operating system.

This document describes the new features related to Version 7.3–1 of the OpenVMS Alpha operating system. For information about how some of the new features might affect your system, read the release notes before you install, upgrade, or use Version 7.3–1.

Document Structure

This manual contains the following parts and chapters:

- Part I, OpenVMS Version 7.3–1 New Features
 - Chapter 1 summarizes the new OpenVMS software features.
 - Chapter 2 provides information about the e-Business technologies that are included in the Compaq OpenVMS e-Business Infrastructure Package.
 - Chapter 3 describes new features of interest to general users of the OpenVMS Alpha operating system.
 - Chapter 4 describes new features that are applicable to the tasks performed by system managers.
 - Chapter 5 describes new features that support programming tasks.
 - Chapter 6 describes significant layered product new features.
- Part II, OpenVMS Documentation
 - Chapter 7 describes the OpenVMS documentation changes from the previous version.
 - Chapter 8 describes how the documentation is delivered.
 - Chapter 9 describes each manual in the OpenVMS documentation set.

Related Documents

For additional information about Compaq *OpenVMS* products and services, access the Compaq website at the following location:

<http://www.openvms.compaq.com/>

Reader's Comments

Compaq welcomes your comments on this manual. Please send comments to either of the following addresses:

Internet	openvmsdoc@compaq.com
Mail	Compaq Computer Corporation OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How to Order Additional Documentation

Visit the following World Wide Web address for information about how to order additional documentation:

<http://www.openvms.compaq.com/>

Conventions

The following conventions are used in this manual:

Ctrl/ <i>x</i>	A sequence such as Ctrl/ <i>x</i> indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 <i>x</i>	A sequence such as PF1 <i>x</i> indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.) In the HTML version of this document, this convention appears as brackets, rather than a box.
...	A horizontal ellipsis in examples indicates one of the following possibilities: <ul style="list-style-type: none">• Additional optional arguments in a statement have been omitted.• The preceding item or items can be repeated one or more times.• Additional parameters, values, or other information can be entered.
.	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
()	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one.
[]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.

	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold text	This typeface represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
<i>italic text</i>	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error <i>number</i>), in command lines (/PRODUCER= <i>name</i>), and in command parameters in text (where <i>dd</i> represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
Monospace text	Monospace type indicates code examples and interactive screen displays. In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.

Part I

OpenVMS Version 7.3-1 New Features

Summary of OpenVMS Alpha Version 7.3–1 New Features

OpenVMS Alpha Version 7.3–1 delivers the highest levels of availability, scalability, flexibility, performance, and security that are required for operating in a 24x365 environment. With more than 20 years of proven reliability, OpenVMS continues to enhance its availability and performance by including new technology in the base operating system and in the OpenVMS Cluster software environment.

OpenVMS Alpha Version 7.3–1 focuses on expanding our OpenVMS Storage Area Network (SAN) offering, improving system availability, providing I/O and SMP performance improvements, and enhanced security. This release also includes enhancements that make it easier to port UNIX applications to OpenVMS.

1.1 Performance Features

OpenVMS Version 7.3–1 focuses on making operating system changes that improve system, SMP, and I/O performance, as well as system scaling, with the ultimate goal of improving customer application performance. OpenVMS Alpha V7.3–1 includes a number of performance enhancements across a range of areas:

- I/O performance
- SMP and general system performance
- System scaling
- OpenVMS Cluster performance
- NUMA enhancements

Application performance is determined by the hardware, operating system, middleware, database, and the application itself. The benefit of the performance changes in the operating system on your application will vary depending on your application's use of the enhanced functions, but many applications will see significant performance benefits! OpenVMS engineers have been working directly with many OpenVMS ISVs and customers on enhancements that have shown substantial performance improvement.

In the Fibre Channel storage area, new features such as distributed interrupts, interrupt coalescing, and reductions in lock usage have improved I/O throughput on Fibre Channel by up to 100% over the previous release.

OpenVMS Alpha V7.3–1 makes low level kernel changes that improve the performance of large SMP systems. Application performance will vary, but many will see significant improvements!

The following information highlights some of the performance features in OpenVMS Alpha Version 7.3–1.

Summary of OpenVMS Alpha Version 7.3–1 New Features

1.1 Performance Features

AST Delivery

The AST queues associated with each process are now synchronized by a process-specific spinlock, instead of the SCHED spinlock. This allows AST delivery to occur in parallel on multiple CPUs and avoids heavy spinlock contention on the SCHED spinlock. Queuing an AST still requires SCHED. However, if the AST being queued is for the current process, you can avoid taking the SCHED spinlock.

MailBox I/O

Previous to OpenVMS Version 7.3–1, all devices used the static MAILBOX spinlock. Now each Mailbox device is synchronized by a mailbox specific spinlock for better performance.

Timer Queue Processing

The Timer Queue was previously a linked list of Timer Queue Entries (TQEs). If the number of TQEs on the Timer Queue became large and there was a heavy load of inserts and cancels, it could take a very long time to insert and remove items from the Timer Queue. The Timer Queue is now in a tree format and performance no longer degrades when the size of the list increases.

PEdriver Checksumming

The algorithm to perform PEdriver checksumming is now more efficient. PEdriver checksumming is performed only if the SYSGEN parameter PEX is set to X.

RMS Global Buffer Locking

Now RMS takes advantage of a change in the Lock Manager's fork lock interface. This interface previously required the IOLOCK8 spinlock to be held when the interface was called. Also, any callbacks by the lock manager were done with IOLOCK8 held. The interface now allows callers to indicate IOLOCK8 is not held and not needed on callbacks, thus avoiding the overhead and contention on IOLOCK8.

Faster SYS\$GETJPI System Service

A new SYS\$GETJPI system service obtains some data for the current process much more efficiently (in some cases without even a change to kernel mode).

Shadowing I/O Completion

I/O post processing for shadowing I/Os will now complete on the same CPU for which the device I/O completes, as opposed to always completing on the primary CPU. If the shadow set has Fast Path devices, I/O post processing will not get moved back to the primary CPU to avoid saturating the primary CPU.

Scaling

To improve system scaling, the following enhancements have been made in OpenVMS Alpha Version 7.3–1:

- Larger WSMAX or BALSETCNT

In V7.3–1, all of the preallocated FRED blocks are removed from S0S1 space. These are now allocated only if needed. This has freed up a vast amount of S0S1 space, which in turn allows the system to have either a larger WSMAX (maximum working set size) or BALSETCNT (number of balance set slots in the system page table). The freed up S0S1 space could also be used for larger nonpaged pool.

Summary of OpenVMS Alpha Version 7.3–1 New Features

1.1 Performance Features

- Replicated Data on NUMA Systems

A subset of OpenVMS read-only data is now replicated on NUMA platforms. This includes static system parameters and a subset of operating system cells such as MMG\$GL_PAGESIZE.

Processor-Specific C RTL

The C RTL now provides an optimized copy of DECC\$SHR (named DECC\$SHR_EV56.EXE) that supports the ability of EV6 processors to schedule byte/word instructions. A second version of DECC\$SHR, built in the usual way, is also provided for other Alpha processor types. The OpenVMS installation procedure checks the processor type and sets a logical to the appropriate DECC\$SHR optimized for that processor type. Customizing DECC\$SHR in this way optimizes performance on EV6 systems.

1.2 Security Features

The following security features in OpenVMS Alpha Version 7.3–1 provide flexibility and simplified software development options for authentication and cryptography:

- Common Data Security Architecture (CDSA)
- Kerberos
- Secure Sockets Library (OpenSSL)
- Authentication and Credentials Management Extensions (ACME)

This manual provides an overview of each security feature as well as references for more information.

1.3 UNIX Portability Features

In addition to providing an Open Source Tools CD-ROM that contains software tools to help you port UNIX applications to OpenVMS Alpha, Version 7.3–1 of the operating system includes internal enhancements to support the porting process and multiplatform environments. Extended File Specifications now includes POSIX-compliant time stamps for file access and modification as well as support for hard links and case sensitivity, which are also supported in RMS, system services, and the Compaq C Run-Time Library. ODS-5 system disks are now supported in OpenVMS Alpha Version 7.3–1. These features, along with the Open Source Tools for OpenVMS, provide a comprehensive porting environment for application developers and an open, flexible computing environment for users and system managers.

1.4 System Management Tools

System managers will be pleased with the new Graphical Configuration Manager (GCM) for viewing and controlling partitioned AlphaServers running OpenVMS. OpenVMS Cluster managers can now override automatic circuit and port selection to optimize performance and availability. Use either the SCACP utility or the Availability Manager to access this new feature.

Summary of OpenVMS Alpha Version 7.3–1 New Features

1.5 Summary Table

1.5 Summary Table

OpenVMS Alpha Version 7.3–1 includes all the capabilities of OpenVMS Alpha Version 7.3 plus the new features added to the OpenVMS Alpha operating system. Table 1–1 summarizes each feature provided by OpenVMS Alpha Version 7.3–1 and presents these features according to their functional component (e-Business, general user, system management, programming, and associated products).

Table 1–1 Summary of OpenVMS Alpha Version 7.3–1 Software Features

Compaq OpenVMS e-Business Infrastructure Package for OpenVMS Alpha	
Compaq OpenVMS e-Business	<p>The following e-Business technologies are included in the Compaq OpenVMS e-Business Infrastructure Package with OpenVMS Alpha:</p> <ul style="list-style-type: none"> • Attunity Connect "On Platform" Package • Compaq BridgeWorks • COM for OpenVMS • Compaq Enterprise Directory for e-Business • NetBeans for OpenVMS • Reliable Transaction Router (RTR) • Compaq Secure Web Browser for OpenVMS Alpha • Compaq Secure Web Server for OpenVMS Alpha • Simple Object Access Protocol (SOAP) Toolkit • Compaq Software Development Kit (SDK) for the OpenVMS Operating System, <i>for the Java™ Platform</i> • Compaq XML Technology <p>The Compaq OpenVMS e-Business Infrastructure Package provides key Internet and e-Business software technology that enhances the base OpenVMS Alpha operating system. These technologies are licensed with the OpenVMS Alpha operating system.</p>
General User Features	
Documentation	<p>Documentation updates are now available on the Internet. Four new manuals are provided with this release. The <i>Guide to Extended File Specifications</i> and the <i>OpenVMS Master Index</i> have been archived.</p>
2D graphics support on 3Dlabs Oxygen VX1-AGP4X graphics module	<p>OpenVMS Alpha now provides support software for this module, including DMA support for both the PCI-based module and AGP-based versions of the VX1 module.</p>
DCL commands and lexical functions	<p>Numerous enhancements have been added to DCL commands and lexical functions to support ODS-5 volumes, RAD functionality, security, and many other features.</p>
WWPPS supports codeset GB18030	<p>The new codeset GB18030-2000 supports simplified Chinese and traditional Chinese.</p>

(continued on next page)

Summary of OpenVMS Alpha Version 7.3–1 New Features 1.5 Summary Table

Table 1–1 (Cont.) Summary of OpenVMS Alpha Version 7.3–1 Software Features

System Management Features	
ACCOUNTING utility	The new /WIDE qualifier changes the width of Buffered I/O and Direct I/O fields to accommodate larger numbers in a report.
AlphaServer DS25	With a processor speed of 1 GHz and support for the latest Gigabit Ethernet adaptor architecture, the new AlphaServer DS25 provides superior performance in a high-powered computing environment. The AlphaServer DS25 supports CD-R and CD-RW drives.
ANALYZE/DISK_STRUCTURE utility	The new /LOCK_VOLUME qualifier lets you prevent file system activity on a volume while you are analyzing that volume.
Authentication and Credentials Management Extensions (ACME)	ACME provides applications with a simple way to provide fully compliant OpenVMS authentication from a single API.
BACKUP utility /DENSITY qualifier	New keywords let you specify the recording density on magnetic tape.
Batch job support for NUMA Resource Affinity Domains	Updates have been made to the batch processing subsystem to support Resource Affinity Domains (RADs) in a NUMA environment.
DECram Version 3.1	DECram V3.1 provides performance improvements, a new command interface, and is supported on Galaxy.
DECram, Mailbox driver, and Volume Shadowing performance gains on multiprocessor systems	DECram for OpenVMS Version 3.1 and Version 2.5, the Mailbox driver, and Volume Shadowing for OpenVMS have been modified to improve I/O performance on multiprocessor systems.
Fast Path performance improvements	Fast Path capable devices have their hardware interrupts targeted to a specific Fast Path CPU, freeing up CPU cycles on the primary CPU.
File system enhancements	POSIX-compliant timestamps, access dates, hard links, and preservation of case sensitivity create a more flexible, interoperable environment for UNIX and OpenVMS.
Graphical Configuration Manager (GCM)	GCM provides a visual means of viewing and controlling the configuration of partitioned AlphaServer systems running OpenVMS.
Kerberos	A full port of MIT's Kerberos 1.0.5 release is now integrated into the OpenVMS operating system. Previously, Kerberos was offered as a layered product.
LANCP	New LANCP qualifiers expand the functionality of the SET DEVICE and SHOW DEVICES commands.
LIBDECOMP.COM	LIBDECOMP.COM can now both expand and reduce selected libraries; it also can list the size and status of all libraries known to the utility. The selection of libraries that ship in data-reduced format has been modified to maximize the benefits of the utility.
ODS-5 volume structure	The legal character set for filenames has been extended to include the ISO Latin-1 Multinational character set (except * and ?). An ODS-5 volume can now be a system disk.

(continued on next page)

Summary of OpenVMS Alpha Version 7.3–1 New Features

1.5 Summary Table

Table 1–1 (Cont.) Summary of OpenVMS Alpha Version 7.3–1 Software Features

System Management Features	
OpenVMS Cluster performance improvements	<p>Many modules that are used to send messages and to transfer data in a cluster have been optimized for speed of execution for the Alpha architecture. Several major modules have been completely rewritten, routine by routine, for performance, correctness, and clarity.</p> <p>Optimizations to the Fibre Channel driver software reduce I/O lock hold time by 3–6 microseconds per I/O, resulting in significant I/O performance improvements.</p> <p>In Fibre Channel configurations, I/O completion interrupts in the host bus adapter are grouped together and then are transmitted as a group instead of being sent one at a time. This new implementation improves I/O performance in environments with a high I/O work load.</p> <p>The limit of 512 devices that could be MSCP served has been increased to 1000. A single OpenVMS Alpha server can now serve up to 1000 disks. This gives system managers greater flexibility in configuring their OpenVMS Cluster storage.</p> <p>Multipath failover to an MSCP-served path for disks is implemented in SCSI and Fibre Channel configurations.</p> <p>Multipath balancing means that the selection of the current path to a multipath disk or tape device is now biased toward the connected path with the fewest devices using it.</p> <p>Multipath tape support, including failover, is implemented for Fibre Channel configurations.</p> <p>Now you can set priority levels for ports and circuits. In addition, PEdriver has been redesigned to update the load class value of its SCS circuits.</p>
OpenVMS Registry	Performance has been improved by indexing the Registry database. There are two performance enhancements to the Registry Control Program.
SHOW CLUSTER utility	New fields have been added to the CIRCUITS class and to the LOCAL_PORTS class to display additional information about virtual circuits and the local system interface.
Subprocess performance improvement	A new method for automatically naming spawned subprocesses reduces overhead.
SYSMAN utility	New RESERVED_MEMORY commands and qualifiers have been added to support RAD functionality.
System parameters	The new parameters DCL_CLTFLAGS, DELPRC_EXIT, MPDEV_AFB_INTVL, and RMS_SEQFILE_WBH support performance options in OpenVMS V7.3–1.
Systems Communications Architecture Control Program (SCACP)	New capabilities that improve flexibility have been added to the SCACP utility that monitors and manages cluster communications.
Ultra3 SCSI Adapter with Fast Path support	KZPEA, a new Ultra3 SCSI adapter, supports Fast Path.
XA capability added to DECdtm	The X/Open Distributed Transaction Processing XA interface has been implemented for OpenVMS. This interface enables the OpenVMS DECdtm transaction manager to coordinate XA-compliant resource managers, and enables the XA-compliant transaction processing systems to coordinate DECdtm-compliant resource managers.

(continued on next page)

Summary of OpenVMS Alpha Version 7.3–1 New Features

1.5 Summary Table

Table 1–1 (Cont.) Summary of OpenVMS Alpha Version 7.3–1 Software Features

Programming Features	
New Alpha processor performance information	New white paper published on the Compaq Web site reveals how programmers can take advantage of the latest performance enhancements on Alpha.
ANALYZE Usage Summary	The new /SHADOW_MEMBER qualifier lets you determine the shadow set member you want to contain the system dump file.
Compaq C Run-Time Library (CRTL) enhancements	Support for files larger than 2-GB files. C RTL features enabled at compile/link time. Enhanced support for UNIX style file names. POSIX root support. 32-bit group identifiers.
DECdtm	Several system services for transaction processing used with DECdtm are now fully documented.
Common Data Security Architecture (CDSA) and Secure Sockets Layer (SSL)	CDSA is an open-source, multiplatform industry standard security architecture. Including CDSA as part of OpenVMS simplifies software development by providing standards based cryptographic interfaces for developers. SSL provides a certificate tool for viewing and creating SSL certificates.
Compaq SSL for OpenVMS Alpha with LDAP	Compaq SSL for OpenVMS Alpha can be established on an LDAP session on a server that supports LDAP.
New behavior of the floating-point registers and execution data block (FRED)	The FRED block is now allocated on an as-needed basis to improve scaling and performance for applications using kernel threads.
Future architecture-specific code changes	Information pertaining to a future version of OpenVMS is provided for the convenience of application developers.
Support for image sections in Galaxy	Write shared image sections can now be placed in Galaxy global sections to extend the application capabilities supported in Galaxy.
Mailbox buffer quotas increased	\$CREMBX system service supports larger mailbox buffer quotas to improve application scaling.
RMS enhancements	An increase in default I/O transfer size reduces I/O for more efficient performance. Support for POSIX-compliant timestamps added. New RMS write-behind performance option enables overlapping I/O, which can be a big performance win for some applications.
System Dump Analyzer	New commands and qualifiers have been added to improve performance, including new commands for analyzing the extended file cache (XFC).
SDA Spinlock Tracing utility	Two new qualifiers, /RATES and /TOTALS, have been added to the SHOW TRACE and SHOW COLLECT commands to enhance your ability to monitor and tune spinlock activity.
System services	Several system services have been added and updated. Updates have been made to system services to support batch jobs for NUMA resource affinity domains (RADs).
Threads libraries supported on NUMA	To improve application scaling and performance on GS Series Alpha (NUMA) systems, a threaded program can now utilize more than one RAD of a NUMA system.

(continued on next page)

Summary of OpenVMS Alpha Version 7.3–1 New Features

1.5 Summary Table

Table 1–1 (Cont.) Summary of OpenVMS Alpha Version 7.3–1 Software Features

Associated Products Features	
Compaq Advanced Server for OpenVMS Alpha	Version 7.3 is supported on OpenVMS Alpha Version 7.3–1.
Compaq C Run-Time Header Files Library	The header files that contain library functions and macro definitions are provided on the OpenVMS Alpha Version 7.3–1 Layered Products CD-ROM.
Compaq PATHWORKS for OpenVMS (Advanced Server)	Version 6.1 is supported on OpenVMS Alpha Version 7.3–1.
Compaq TCP/IP Services for OpenVMS	Version 5.3 of this product is supported on OpenVMS Alpha Version 7.3–1.
OpenVMS Management Station Version 3.2	OpenVMS Management Station Version 3.2 is included with OpenVMS Alpha Version 7.3–1.
Powerstorm 300/350 Graphics support	Version 2.0 includes new features for the OpenGL Utility Toolkit (GLUT).
UNIX Portability Features	UNIX portability features are introduced for easier porting of UNIX applications to OpenVMS.
Open Source Tools for OpenVMS	The Open Source Tools CD-ROM, included with OpenVMS Version 7.3–1, has a number of utilities and sources to expedite porting from UNIX to OpenVMS.

Compaq OpenVMS e-Business Features

This chapter provides information about the e-Business technologies that are included in the Compaq OpenVMS e-Business Infrastructure Package with OpenVMS Alpha Version 7.3-1. This package provides key Internet and e-Business software technology that enhances the base OpenVMS Alpha operating system. These technologies are licensed with the OpenVMS Alpha operating system.

The Compaq OpenVMS e-Business Infrastructure Package contains the following software and accompanying documentation:

- Attunity Connect "On Platform" Package
- Compaq BridgeWorks
- COM for OpenVMS
- Compaq Enterprise Directory for e-Business
- NetBeans for OpenVMS
- Reliable Transaction Router (RTR)
- Compaq Secure Web Browser for OpenVMS Alpha (based on Mozilla)
- Compaq Secure Web Server for OpenVMS Alpha (based on Apache)
- Simple Object Access Protocol (SOAP) Toolkit
- Compaq Software Development Kit (SDK) for the OpenVMS Operating System, *for the Java™ Platform*
- Compaq XML (Extensible Markup Language) Technology

Refer to the *Compaq OpenVMS e-Business Infrastructure Package CD-ROM Booklet* and the *Compaq OpenVMS e-Business Infrastructure Package Software Product Description, 80.58.xx* included in the e-Business package for more detailed information.

For up-to-date information on OpenVMS e-Business technologies, refer to the following Web site:

<http://www.openvms.compaq.com/ebusiness/>

The following sections briefly describe the e-Business software and provide pointers and Web sites for further information. Refer to the *Compaq OpenVMS e-Business Infrastructure Package SPD* for technology descriptions, other software requirements, and licensing information. The technology on the e-Business CD-ROM has been tested and qualified with OpenVMS Alpha Version 7.2-2 and later.

Compaq OpenVMS e-Business Features

2.1 Attunity Connect "On Platform" Package

2.1 Attunity Connect "On Platform" Package

Attunity Connect is object-oriented middleware that facilitates the development of applications that access, integrate, and update data from multiple, heterogeneous sources across a wide range of operating system platforms. With Attunity Connect, you can extend the life of your existing data and applications and preserve your significant IT investments.

The e-Business CD-ROM contains the Attunity Connect "On Platform" Package for OpenVMS Alpha. You can also download the Attunity Connect "On Platform" Package from the following OpenVMS Web site:

<http://www.openvms.compaq.com/openvms/products/ips/attunity/>

2.2 Compaq BridgeWorks

Compaq BridgeWorks is an automated component creation tool that produces ready-to-build, distributable components from your existing OpenVMS applications. By creating standard reusable components, you extend the reach of your OpenVMS applications so that they can serve many of today's Web- and desktop-based clients.

Compaq BridgeWorks provides a quick and reliable way to integrate OpenVMS applications on Alpha or VAX systems with the world of components and distributed objects. Using Compaq BridgeWorks, you can readily reuse your applications with some of the latest technologies, making them accessible from the desktops and the Web browsers of users while maintaining the high availability of the OpenVMS operating system.

For more information about Compaq BridgeWorks, refer to the *Compaq OpenVMS e-Business Infrastructure Package Software Product Description*, or go to the Compaq BridgeWorks Web site:

<http://www.openvms.compaq.com/commercial/bridgeworks/>

2.3 COM Version 1.2 for OpenVMS

Component Object Model (COM) is a technology from Microsoft that allows developers to create distributed network objects. Digital Equipment Corporation and Microsoft jointly developed the COM specification. COM was first released as NetOLE (Network OLE), renamed DCOM (Distributed COM), and then renamed COM for OpenVMS.

OpenVMS Version 7.3-1 includes COM for OpenVMS Version 1.2. This version of COM has upgraded the source base to Windows NT 4.0 Service Pack 5. Previous versions of COM for OpenVMS were based on SP3.

COM Version 1.2 for OpenVMS has removed all C compiler requirements for building applications, and provides the ability to run components in DLL surrogate processes. COM Version 1.2 has also fixed several problems found in COM Version 1.1-B. For more detailed information, refer to the *COM, Registry, and Events for OpenVMS Developer's Guide*.

The Compaq COM for OpenVMS kit included on the e-Business CD-ROM provides all the code and documentation you need to install Compaq COM for OpenVMS on your system and to develop COM applications. This product is also available to download from the following Web site:

<http://www.openvms.compaq.com/openvms/products/dcom/>

2.4 Compaq Enterprise Directory for e-Business

Compaq OpenVMS Enterprise Directory for e-Business combines the best of both industry standard LDAPv3 and X.500 capabilities, and delivers robust and scalable directory services across intranets, extranets, and the Internet to customers, suppliers, and partners. Lightweight Directory Access Protocol (LDAP) support allows access by a myriad of LDAP-based clients, user agents, and applications. The X.500 support brings high performance, resilience, advanced access controls, and easy replication across the enterprise. This latest release includes support for SSL/TLS between clients and directory servers, Selective Shadowing of data to consumer directories, and a new Java based Management GUI.

For further information, refer to the *Compaq OpenVMS Enterprise Directory for e-Business Software Product Description (SPD 81.03.01)* included on the e-Business CD-ROM in the Enterprise Directory Services documentation directory. This product is also available to download from the following Web site:

<http://www.openvms.compaq.com/commercial/edir/>

2.5 NetBeans for OpenVMS

NetBeans provides an open-source, modular, and integrated development environment (IDE) for Java and JavaBeans development implemented in pure Java™. The popularity of NetBeans is a result of its versatility, extensible architecture, and relative ease of use. NetBeans for OpenVMS is included on the e-Business CD-ROM. This product is also available to download from the following Web site:

<http://www.openvms.compaq.com/openvms/products/ips/netbeans/>

2.6 Reliable Transaction Router (RTR)

Reliable Transaction Router (RTR) is fault tolerant transactional messaging middleware used to implement large, distributed applications using client/server technology. Reliable Transaction Router enables computing enterprises to deploy distributed applications on OpenVMS Alpha and VAX systems.

Refer to the *Reliable Transaction Router for OpenVMS Software Product Description (SPD 51.04.xx)* included on the e-Business CD-ROM for additional information; or you can access the RTR Web site at:

<http://www.compaq.com/rtr/>

2.7 Compaq Secure Web Browser for OpenVMS Alpha (based on Mozilla)

Compaq Secure Web Browser (CSWB) is the officially supported Web browser for OpenVMS Alpha, based on the Mozilla Milestone 1.0 Release Candidate 2. The Mozilla open-source project (<http://mozilla.org/>) was started in 1998 by Netscape Communications Corporation. The Mozilla Web Browser is designed for standards compliance, performance and portability. Mozilla uses the same code base as Netscape 6.x.

Compaq OpenVMS e-Business Features

2.7 Compaq Secure Web Browser for OpenVMS Alpha (based on Mozilla)

CSWB includes support for the following features:

- HyperText Markup Language (HTML 4.01)
- Cascading Style Sheets (CSS1/2)
- Document Object Model (DOM1/2)
- eXtensible Markup Language (XML)
- Resource Definition Framework (RDF)
- Secure Socket Layer (SSL)
- JavaScript
- Java

For more information about CSWB, refer to the *Compaq Secure Web Browser for OpenVMS Alpha Installation Guide and Release Notes*. This product is also available to download from the CSWB Web site:

<http://www.openvms.compaq.com/openvms/products/ips/cswb/cswb.html>

2.8 Compaq Secure Web Server for OpenVMS Alpha (based on Apache)

Compaq Secure Web Server for OpenVMS Alpha (CSWS) is based on the popular Apache Web Server from the Apache Software Foundation. Building on the source code from the Apache Software Foundation (<http://www.apache.org>), Compaq OpenVMS engineering has incorporated and fully integrated OpenSSL with `mod_ssl`, the most popular open-source implementations of SSL.

Compaq Secure Web Server for OpenVMS Alpha Version 1.2 is supplied by Compaq Computer Corporation under the terms of the open source license from the Apache Software Foundation, and is based on the Version 1.3.20 release from the Apache Software Foundation. New features in Version 1.2 include two new modules, `mod_proxy` and `mod_rewrite`, and support for clusterwide shared memory on OpenVMS Galaxy system clusters. Version 1.2 also includes a new optional kit, `CSWS_PHP`.

For more detailed information, refer to the *Compaq Secure Web Server Installation and Configuration Guide and Release Notes* and see the CSWS Web site at:

<http://www.openvms.compaq.com/openvms/products/ips/apache/csWS.html>

From the CSWS home page, you can download the following optional CSWS kits:

CSWS_JAVA
CSWS_PERL
CSWS_PHP

2.9 Simple Object Access Protocol (SOAP) Toolkit

SOAP provides a simple, lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment. SOAP is an XML-based protocol that consists of three parts: an envelope that defines a framework for describing the contents of a message and how to process it, a set of encoding rules for expressing application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP defines

Compaq OpenVMS e-Business Features

2.9 Simple Object Access Protocol (SOAP) Toolkit

a simple mechanism for expressing application semantics that allows SOAP to be used in a wide variety of systems. SOAP is included on the e-Business CD-ROM.

This product is also available to download from the following Web site:

<http://www.openvms.compaq.com/openvms/products/ips/soap/>

2.10 Compaq Software Development Kit (SDK) for the Java™ Platform

The Compaq SDK provides an environment in which to develop and deploy Java applications on OpenVMS Alpha. Java applications can be written once and run on any operating system that implements the Java run-time environment, which consists primarily of the Java Virtual Machine (JVM).

The Compaq SDK now includes the Compaq Fast VM, a new Just-In-Time (JIT) compiler technology designed to provide optimal Java run-time performance on OpenVMS Alpha systems. Fast VM offers significant performance advantages over the Classic JIT provided with the Compaq Software Development Kit (SDK).

The Compaq Software Development Kit (SDK) for the OpenVMS System, *for the Java™ Platform* is included on the e-Business CD-ROM, or you can download this kit from the Compaq Java home page at the following Web address:

<http://www.compaq.com/java/download/index.html>

2.11 Compaq XML Technology

The following components are provided on the e-Business CD-ROM using open source software from the Apache Software Foundation:

- XML parsers in Java and C++
- XSLT style sheet processors in Java and C++

This technology provides applications the ability to parse, generate, manipulate, validate, and transform Extensible Markup Language (XML) documents and data. This product is also available to download from the following Web site:

<http://www.openvms.compaq.com/openvms/products/ips/xml/>

General User Features

This chapter provides information about new features for all users of the Compaq OpenVMS operating system.

3.1 Documentation Updates on the World Wide Web

In an effort to provide timely updates for our customers, OpenVMS Documentation will post new, supplemental, and updated information to all OpenVMS product documentation sets on the World Wide Web. We encourage customers to check the OpenVMS Documentation web site for changes to the OpenVMS and layered product manuals, as well as pertinent papers and other documents of interest to our customers. Documents posted on the site will be clearly marked and will include the revision date.

Our Web site is located at the following URL:

<http://www.openvms.compaq.com/doc>

3.2 New and Archived Manuals

The following new manuals are now included in the OpenVMS documentation set:

- *COM, Registry, and Events for OpenVMS Developer's Guide*
- *Open Source Security for OpenVMS Alpha, Volume 1: Common Data Security Architecture*
- *Open Source Security for OpenVMS Alpha, Volume 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha*
- *Compaq C Run-Time Library Reference Manual for OpenVMS Systems*

The following manuals have been archived:

- *OpenVMS Master Index*
- *OpenVMS Guide to Extended File Specifications*

The information contained in *OpenVMS Guide to Extended File Specifications* has been distributed to the following manuals in the OpenVMS documentation set:

- *OpenVMS User's Manual*
- *Guide to OpenVMS File Applications*
- *OpenVMS System Services Reference Manual*
- *OpenVMS Programming Concepts Manual*

Refer to Part II for more information.

General User Features

3.3 3Dlabs Oxygen VX1-AGP4X 2D Graphics Support

3.3 3Dlabs Oxygen VX1-AGP4X 2D Graphics Support

OpenVMS Alpha Version 7.3–1 incorporates 2D support for the 3Dlabs Oxygen VX1-AGP4X graphics module. This module can be plugged into an AGP slot on a supported platform (for example, the AlphaServer ES45 Model 1B). To learn which platforms offer AGP device support, ask a Compaq customer service representative or refer to the following Web site:

<http://www.compaq.com/alphaserver/technology/index.html>.

2D support of the 3Dlabs Oxygen VX1-PCI graphics module, introduced with OpenVMS Alpha Version 7.3, continues to be supported in OpenVMS Alpha Version 7.3–1. AlphaServer ES45 users with multihead configurations now have the choice of using only VX1-PCI modules or of using one or more VX1-PCI modules in combination with a VX1-AGP module.

For details about using the 3Dlabs Oxygen VX1-AGP4X graphics module, refer to the *3Dlabs OXYGEN VX1 PCI and AGP Graphics Controller Installation Guide*, order number EK-VX1GC-IG.

3.4 DCL Commands and Lexical Functions

This section describes new and changed DCL commands, qualifiers, and lexical functions for OpenVMS Version 7.3–1. The following tables summarize these changes.

For more information, refer to the *OpenVMS DCL Dictionary*.

DCL Command	Documentation Update
ANALYZE/PROCESS	A restriction note has been added to the /IMAGE_PATH qualifier.
COPY/FTP	A new qualifier, /PASSIVE, has been added.
CREATE/DIRECTORY	The /ALLOCATE qualifier has been updated to include ODS-5 files.
DELETE	A new qualifier, /IGNORE=INTERLOCK, has been added.
INITIALIZE	The following tape-density keywords have been added to the /DENSITY qualifier: AIT2, AIT3, AIT4, DLT8000, 89000, SDLT, and SDLT320.
INITIALIZE/QUEUE	A new qualifier, /RAD, has been added, along with an example.
MOUNT	The following tape-density keywords have been added to the /DENSITY qualifier: AIT2, AIT3, AIT4, DLT8000, 89000, SDLT, and SDLT320.
PURGE	A new qualifier, /IGNORE=INTERLOCK, has been added.
READ	The Description section has been updated with new information about network block count.
RECALL/ALL	A new parameter has been added, along with a new example.
SET AUDIT	The /ENABLE qualifier text has been updated, and DECnet-Plus has been added to the CONNECTION keyword list.

General User Features

3.4 DCL Commands and Lexical Functions

DCL Command	Documentation Update
SET COMMAND	A note has been added to the /OBJECT qualifier to describe the use of the CDU\$TABLE_SPACE logical name that lets you increase the value of virtual memory.
SET CPU	The following new qualifiers have been added: /AUTO_START, /FAILOVER, /MIGRATE, /OVERRIDE_CHECKS, /POWER, /REFRESH, /START
SET ENTRY	A new qualifier, /RAD, has been added, along with an example.
SET MAGTAPE	The following tape-density keywords have been added to the /DENSITY qualifier: AIT2, AIT3, AIT4, DLT8000, 89000, SDLT, and SDLT320.
SET QUEUE	A new qualifier, /RAD, has been added, along with an example.
SET SERVER	Support for ACME, including new qualifiers, has been added.
SET VOLUME	The ODS-5 system disk restriction has been removed.
SHOW/CPU	The following new qualifiers have been added: /CONFIGURE_SET, /POTENTIAL_SET, /STANDBY_SET, /SYSTEM
SHOW DEVICES	A new qualifier, /SIZE, has been added to display free space in bytes, along with an example.
SHOW ENTRY	This command has been enhanced to display RAD information.
SHOW NETWORK	New examples have been added.
SHOW QUEUE/FULL	This command has been enhanced to display RAD information.
SHOW SERVER	Support for ACME, including new qualifiers and examples, has been added.
SPAWN	A note has been added to the /PROCESS qualifier regarding the use of the DCL_CTLFLAGS system parameter.
START/CPU	A new qualifier, /POWER, has been added.
START/QUEUE	A new qualifier, /RAD, has been added, along with an example.
STOP	Two new qualifiers, /EXIT and /IMAGE, have been added.
STOP/CPU	Two new qualifiers, /MIGRATE and /POWER, have been added.
SUBMIT	A new qualifier, /RAD, has been added, along with an example.

DCL Lexical	Documentation Update
F\$CVTIME	The following keywords have been added, along with an example: DAYOFYEAR, HOUROFYEAR, MINUTEOFYEAR, SECONDOFYEAR
F\$GETDVI	Two new item codes, MPDEV_CURRENT_PATH and MULTIPATH, have been added.

General User Features

3.4 DCL Commands and Lexical Functions

DCL Lexical	Documentation Update
F\$GETJPI	The following items have been added: CLASSIFICATION, CURRENT_CAP_MASK, IMAGE_AUTHPRIV, IMAGE_PERMPRIV, IMAGE_WORKPRIV, INSTALL_RIGHTS, INSTALL_RIGHTS_SIZE, PERMANENT_CAP_MASK, PERSONA_AUTHPRIV, PERSONA_ID, PERSONA_PERMPRIV, PERSONA_RIGHTS, PERSONA_RIGHTS_SIZE, PERSONA_WORKPRIV, SCHED_CLASS_NAME, SUBSYSTEM_RIGHTS, SUBSYSTEM_RIGHTS_SIZE, SYSTEM_RIGHTS_SIZE
F\$GETQUI	A new example for deleting retained entries from queues has been added. A restriction has been added to the Description section regarding the use of wildcards and DCL queue-related commands.
F\$GETSYI	The following items have been added: ACTIVE_CPU_MASK, AVAIL_CPU_MASK, COMMUNITY_ID, CPU_AUTOSTART, CPU_FAILOVER, CPUCAP_MASK, GALAXY_ID, GALAXY_MEMBER, GALAXY_PLATFORM, GALAXY_SHMEMSIZE, GLX_FORMATION, GLX_MAX_MEMBERS, GLX_MBR_MEMBER, GLX_MBR_NAME, GLX_TERMINATION, HP_ACTIVE_CPU_CNT, HP_ACTIVE_SP_CNT, HP_CONFIG_SBB_CNT, HP_CONFIG_SP_CNT, MAX_CPUS, PARTITION_ID, POTENTIAL_CPU_MASK, POTENTIALCPU_CNT, POWERED_CPU_MASK, POWEREDCPU_CNT, PRESENT_CPU_MASK, PRESENTCPU_CNT, PRIMARY_CPUID, SCSNODE

For information on the DCL commands and lexicals related to case sensitivity, hard links, and access dates, see Section 4.10.

3.4.1 RECALL/ALL Update

The RECALL/ALL command now accepts a parameter that allows you to display multiple occurrences of an individual command within the recall buffer.

For example:

```
$ RECALL/ALL SHOW
 1 show system
 4 show users
 5 show intrusion
 6 show error
 7 show time
```

3.4.2 SET COMMAND/OBJECT Update

You can now use the new logical, CDU\$TABLE_SPACE, to increase the value of virtual memory allocated by the SET COMMAND/OBJECT command.

Prior to Version 7.3–1, when compiling very large command definition (CLD) files, you may have encountered the following error:

```
%CDU-F-INTNODESPACE, Internal error: node space exhausted
```

The default value of virtual memory allocated by SET COMMAND/OBJECT is 384KB. This value may not be sufficient to process large CLD files.

General User Features

3.4 DCL Commands and Lexical Functions

To increase this size, use the `CDU$TABLE_SPACE` logical name to specify a larger value and reissue the command. For example:

```
$ DEFINE CDU$TABLE_SPACE 2048
$ SET COMMAND/OBJECT MYCLD.CLD
$ DIR *.OBJ

Directory $1$DKA600:[GUY.CDU]
MYCLD.OBJ;1
```

3.4.3 SHOW DEVICES Update

You can now use the `SHOW DEVICES` command to display free space in bytes to accommodate large disk sizes.

Prior to Version 7.3–1, the `SHOW DEVICES` command displayed free space only in blocks.

Beginning with Version 7.3–1, you can use the `SHOW DEVICES/SIZE=BYTES` command to display free space in bytes. For example:

```
$ SHOW DEVICES DKB/SIZE=BYTES

Device          Device      Error   Volume      Free  Trans  Mnt
Name            Status     Count   Label       Space Count  Cnt
$2$DKB0:        (IPG32)   Mounted 0   ENG_IMAGES  5.41GB   1   3
$2$DKB100:      (IPG32)   Online  0
$2$DKB300:      (IPG32)   Online  0
$2$DKB400:      (IPG32)   Online  0   (remote shadow member)
$2$DKB500:      (IPG32)   Mounted 0   GUY         621.37MB 1   4
$2$DKB600:      (IPG32)   Mounted 0   DATA       11.54GB  1   4
$
```

Note that the `SHOW DEVICES/FULL` command still displays free space in blocks.

3.5 WWPPS Adds Support for GB18030 Codeset

The World-Wide PostScript Printing Subsystem (WWPPS) now supports the new, compulsory standard, codeset GB18030-2000 for simplified Chinese and traditional Chinese. Support for codeset GB18030 is required for all vendors who want to do business in the People's Republic of China (PRC).

For more information about WWPPS, refer to the *OpenVMS User's Manual*.

System Management Features

This chapter provides information about new features, changes, and enhancements for system managers.

4.1 ACCOUNTING Utility Has New Qualifier

The /WIDE qualifier in the ACCOUNTING utility changes the width of Buffered I/O and Direct I/O fields in a report from 8 to 10 characters. Without the /WIDE qualifier, these fields print asterisks (*****), if the field overflows.

For more information, refer to the *OpenVMS System Management Utilities Reference Manual*.

4.2 AlphaServer DS25

OpenVMS Alpha Version 7.3–1 supports the AlphaServer DS25, the newest addition to the AlphaServer DS20 family. With a processor speed of 1 GHz and support for the latest Gigabit Ethernet adaptor architecture, the AlphaServer DS25 provides superior performance in a high-powered computing environment.

4.2.1 Broadcom 5700 Series Gigabit Ethernet

The AlphaServer DS25 supports the 3COM Gigabit Ethernet adaptor (DEGXA) as a LAN device and a cluster interconnect device. This support will be included in future AlphaServers as well.

4.2.2 Use New CDRECORD.COM to Create a CD-ROM

Read and write-once support for CD-R and CD-RW drives is introduced in OpenVMS Alpha Version 7.3–1 on the AlphaServer DS25 system. On these systems, and future AlphaServer systems as well, you can create your own CD-ROMs for distributing or backing up files.

To create a CD-ROM, you need a CD-Recordable (CD-R or CD-RW) drive and a blank CD-R disk. CD-R and CD-RW drives use a laser beam to write (or burn) data to a blank CD-R disk. A CD-R disk is “write once.” This means you can write data on it one time only. It is not rewritable.

The write process creates a CD-ROM in Files-11 format. Any supported CD-ROM reader on a computer running OpenVMS will be able to read the CD-ROMs you create.

Note

You can create a CD-ROM that contains data files, but audio recording is not supported at this time.

System Management Features

4.2 AlphaServer DS25

OpenVMS supports only qualified CD-R and CD-RW drives. For more information on Alpha systems and the drives they support, refer to the appropriate page at the following Web site:

<http://www.compaq.com/alphaserver/configure.html>

You cannot use COPY commands to transfer files from a hard drive to a CD-R disk. You use a program called CDRECORD.COM.¹ The sources for CDRECORD.COM are provided on the Open Source Tools for OpenVMS CD-ROM (see Section 6.8.) For more detailed information on using CDRECORD.COM, refer to the *OpenVMS System Manager's Manual, Volume 1: Essentials*.

4.3 ANALYZE/DISK_STRUCTURE Utility Has New Qualifier

The /LOCK_VOLUME qualifier in the ANALYZE/DISK_STRUCTURE utility provides a way to prevent file system activity on a volume while you are analyzing that volume with the ANALYZE/DISK_STRUCTURE utility.

For more information, refer to the *OpenVMS System Management Utilities Reference Manual*.

4.4 Authentication and Credentials Management Extensions (ACME) Subsystem

The Authentication and Credentials Management Extensions (ACME) subsystem provides authentication and persona-based credential services.

Applications can use these services to interact with the user to perform one or more of the following functions:

- User authentication
- Password change
- Persona creation and modification

ACME supports standard OpenVMS authentication and external authentication policies; therefore, applications utilize the same mechanisms as used by the system's LOGINOUT and SET PASSWORD components.

For general information on the ACME subsystem, refer to *OpenVMS Guide to System Security*.

Information about the ACME subsystem components is located in the following manuals:

ACME Component	Documentation Location
SYS\$ACM system service	<i>OpenVMS System Services Reference Manual</i> and <i>OpenVMS Programming Concepts Manual</i>
ACME_SERVER process and ACME policy-provider agents	<i>OpenVMS Guide to System Security</i>
SET and SHOW SERVER ACME DCL commands	<i>OpenVMS DCL Dictionary</i>

¹ The CDRECORD software internals were developed outside of Compaq, and are protected by the GNU General Public License Version 2. A copy of the sources and the GNU license are on the Compaq OpenVMS Freeware Web page: <http://www.openvms.compaq.com/openvms/freeware/>

4.5 BACKUP Utility Supports New Keywords for /DENSITY Qualifier

The BACKUP command /DENSITY qualifier allows you to specify the recording density of an output magnetic tape. The following table shows the new /DENSITY keywords that are supported beginning in OpenVMS Version 7.3–1:

Keyword	Description
AIT1	Sony Advanced Intelligent Tape 1 - Alpha only
AIT2	Sony Advanced Intelligent Tape 2 - Alpha only
AIT3	Sony Advanced Intelligent Tape 3 - Alpha only
AIT4	Sony Advanced Intelligent Tape 4 - Alpha only
DLT8000	DLT 8000 - Alpha only
8900	Exabyte 8900 - Alpha only
SDLT	SuperDLT1 - Alpha only
SDLT320	SuperDLT320 - Alpha only

Refer to the *OpenVMS System Management Utilities Reference Manual* for more information.

4.6 Batch Job Support for NUMA Resource Affinity Domains

This section describes updates to the OpenVMS batch processing subsystem in support of Resource Affinity Domains (RADs) in a NUMA environment.

System managers now have the ability to assign batch queues to a RAD, and users can specify a RAD on which to run a batch job.

These new features are restricted for use on batch execution queues and batch jobs.

For information about the system services updates to \$GETQUI and \$SNDJBC, see Section 5.15. For information about DCL commands, refer to the *OpenVMS DCL Dictionary*.

4.6.1 Batch Queue Level RAD Support

A new qualifier, /RAD, is available for use with the following DCL commands: INITIALIZE/QUEUE, SET/QUEUE, and START/QUEUE. The system manager specifies the RAD number on which to run batch jobs assigned to the queue.

The RAD value is validated as a positive integer between 0 and SYI\$_RAD_MAX_RADS. The SHOW/QUEUE/FULL command now displays the RAD in its output, and the F\$GETQUI lexical function now accepts a new RAD item.

Examples

This section describes a sequence of commands and their effects on a batch queue. A SHOW command is included in each example to illustrate the batch queue modifications.

- The following INITIALIZE/QUEUE command creates or reinitializes the batch queue BATCHQ1 to run on node QUEBID. All jobs assigned to this queue will run on RAD 0:

System Management Features

4.6 Batch Job Support for NUMA Resource Affinity Domains

```
$ INITIALIZE/QUEUE/ON=QUEBID::/BATCH/RAD=0 BATCHQ1
$ SHOW QUEUE/FULL BATCHQ1
Batch queue BATCHQ1, stopped, QUEBID::
  /BASE_PRIORITY=4 /JOB_LIMIT=1 /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
  /RAD=0
```

- The following **START/QUEUE** command modifies **BATCHQ1** to run all assigned jobs on **RAD 1** of **QUEBID**, and readies the queue to accept jobs for processing:

```
$ START/QUEUE/RAD=1 BATCHQ1
$ SHOW QUEUE/FULL BATCHQ1
Batch queue BATCHQ1, idle, on QUEBID::
  /BASE_PRIORITY=4 /JOB_LIMIT=3 /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
  /RAD=1
```

- The following **SET/QUEUE** command modifies the batch queue to run all assigned jobs on **RAD 0** of **QUEBID**. Any new jobs assigned to the queue will run on **RAD 0**. Jobs already executing on the queue will continue to completion, executing on the previous **RAD** value.

```
$ SET/QUEUE/RAD=0 BATCHQ1
$ SHOW QUEUE/FULL BATCHQ1
Batch queue BATCHQ1, idle, on QUEBID::
  /BASE_PRIORITY=4 /JOB_LIMIT=3 /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
  /RAD=0
```

- The following **SET/QUEUE/NORAD** command erases the **RAD** value for **BATCHQ1**:

```
$ SET/QUEUE/NORAD BATCHQ1
$ SHOW QUEUE/FULL BATCHQ1
Batch queue BATCHQ1, idle, on QUEBID::
  /BASE_PRIORITY=4 /JOB_LIMIT=3 /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
```

- The following **F\$GETQUI** lexical function returns the value of the **RAD**. A value of **-1** that indicates no **RAD** value is attributed to the queue:

```
$ WRITE SYS$OUTPUT F$GETQUI("DISPLAY_QUEUE", "RAD", "BATCHQ1")
-1
```

4.6.2 Job-Level RAD Support

The new qualifier **/RAD** is added for use with the following commands: **SUBMIT** and **SET ENTRY**.

The user specifies in the qualifier value the **RAD** number on which the submitted batch job is to execute. The **SHOW ENTRY** and **SHOW QUEUE/FULL** commands are enhanced to list the **RAD** setting on batch jobs.

Examples

When a job is submitted to a batch queue that does not have a **RAD** setting, the job will execute using the **RAD** specified on the **SUBMIT** command.

The following command submits **TEST.COM** to queue **ANYRADQ**. There is no **RAD** setting on queue **ANYRADQ**:

System Management Features

4.6 Batch Job Support for NUMA Resource Affinity Domains

```
$ SUBMIT/HOLD/QUEUE=ANYRADQ /RAD=1 TEST.COM
Job TEST (queue ANYRADQ, entry 23) holding

$ SHOW ENTRY/FULL 23
  Entry  Jobname      Username      Blocks  Status
  -----
    23   TEST          SYSTEM              Holding
        On idle batch queue ANYRADQ
        Submitted 24-JUL-2001 14:19:37.44 /KEEP /NOPRINT /PRIORITY=100 /RAD=1
        File: _$1$DKB200:[SWEENEY.CLIUTL]TEST.COM;1
```

When a job is submitted to a batch queue that does have a RAD setting, the job executes using the RAD specified on the queue, regardless of the RAD specified on the SUBMIT command. This behavior is consistent with other batch system features.

The queue, BATCHQ1, is defined with /RAD=0. The following SUBMIT command creates a job that runs on RAD 0, even though the user specified RAD 1 on the submission:

```
$ SUBMIT/HOLD/QUEUE=BATCHQ1 /RAD=1 TEST.COM
Job TEST (queue BATCHQ1, entry 24) holding

$ SHOW ENTRY 24/FULL
  Entry  Jobname      Username      Blocks  Status
  -----
    24   TEST          SYSTEM              Holding
        On idle batch queue BATCHQ1
        Submitted 24-JUL-2001 14:23:10.37 /KEEP /NOPRINT /PRIORITY=100 /RAD=0
        File: _$1$DKB200:[SWEENEY.CLIUTL]TEST.COM;2
```

4.6.3 Run-Time Behavior

When you specify a RAD on a batch job, the job controller creates the process with the \$CREPRC **home_rad** argument set to the RAD value on the job.

If the RAD specified on the job is invalid on the target system, the job fails to execute and the job controller sends a message to the operator console indicating that a bad RAD was specified. If the bad RAD value matches the RAD setting on the batch queue, the batch queue is stopped and the job remains in the queue.

4.6.3.1 Error Processing

The following is an example of run-time error processing:

```
SYSTEM@QUEBID> SUBMIT/NONOTIFY/NOLOG/QUEUE=BATCHQ1 TEST.COM
Job TEST (queue BATCHQ1, entry 30) started on BATCHQ1

OPCOM MESSAGES

SYSTEM@QUEBID> START/QUEUE BATCHQ1
%%%%%%%%%% OPCOM 25-JUL-2001 16:15:48.52 %%%%%%%%%%%
Message from user SYSTEM on QUEBID
%JBC-E-FAILCREPRC, job controller could not create a process

%%%%%%%%%% OPCOM 25-JUL-2001 16:15:48.53 %%%%%%%%%%%
Message from user SYSTEM on QUEBID
-SYSTEM-E-BADRAD, bad RAD specified

%%%%%%%%%% OPCOM 25-JUL-2001 16:15:48.54 %%%%%%%%%%%
Message from user SYSTEM on QUEBID
%QMAN-E-CREPRCSTOP, failed to create a batch process,
queue BATCHQ1 will be stopped

$SYSTEM@QUEBID> WRITE SYS$OUTPUT -
_ $ F$message(%x'F$GETQUI("DISPLAY_ENTRY", "CONDITION_VECTOR", "30")')
%SYSTEM-E-BADRAD, bad RAD specified
```

System Management Features

4.6 Batch Job Support for NUMA Resource Affinity Domains

4.6.3.2 RAD Modifications on Batch Queues

When you change the RAD value on a batch queue, the jobs currently in the batch queue are not dynamically updated with the new RAD value specified on the queue.

Any executing jobs will complete processing using the original RAD value. Jobs in the pending, holding, or timed execution state will retain the old RAD value on the job; however, when such a job becomes executable, the job is updated with the new RAD value and runs on the RAD specified on the queue.

4.7 DECram Support for Galaxy

DECram is a disk device driver that enables system managers to create logical disks in memory to improve I/O performance. Data on an in-memory DECram disk can be accessed at a faster rate than data on hardware disks. DECram Version 3.1 runs on OpenVMS Alpha systems, including Galaxy configurations, running OpenVMS Alpha Version 7.2-1H1 or higher.

DECram Version 3.1 takes advantage of the advanced capabilities of the newest Alpha systems, such as Galaxy shared memory. Additionally, the change to I/O post processing described in Section 4.8 is a significant performance enhancement. Version 3.1 allocates memory across Resource Affinity Domains (RADs). It moves the virtual device addressing from the S1 address space into the S2 address space, allowing for the creation and addressing of devices larger than 2 gigabytes (GBs).

You can easily upgrade an Alpha-based system to DECram Version 3.1. With Version 3.1, you can either use the improved new DECram command interface or continue using the familiar SYSMAN commands for creating, initializing, and mounting DECram disks.

DECram Version 3.1 disks are created and formatted at the DECram> prompt and are initialized with the DCL INITIALIZE (INIT) command. If you are configuring DECram Version 3.1 on OpenVMS Alpha, you can generate a DECram startup procedure to set up the disk and copy any required files to it. Usually, this procedure is called from the system startup procedure SYS\$MANAGER:SYSTARTUP_VMS.COM.

Requirement

A DECram disk must have 516 bytes of free page list per block (512 bytes) of disk space allocated.

DECram Version 3.1 and supporting documentation are included on the Software Products Library and the Online Documentation Library CD-ROMs.

4.8 DECram, Mailbox Driver, and Volume Shadowing Performance Gains

DECram for OpenVMS Version 3.1 and Version 2.5, the Mailbox driver, and Volume Shadowing for OpenVMS have been modified to improve performance on multiprocessor systems. A change has been made in OpenVMS Alpha Version 7.3-1 that enables device drivers to specify that I/O post processing be done on the local CPU. DECram, Mailbox driver, and Volume Shadowing have all been modified to use this performance enhancement. The performance gains are greatest for DECram and can be significant for Mailbox driver. For configurations

4.8 DECram, Mailbox Driver, and Volume Shadowing Performance Gains

that use Volume Shadowing, if the primary CPU is saturated, such systems will derive modest benefits.

Prior to the introduction of this capability, two routines were called to complete an input-output (I/O) operation, IOC\$REQCOM and IOC\$ALTREQCOM. These routines cause non-fast-path I/O completion to be performed by the primary CPU, thus potentially degrading performance.

As of Version 7.3–1, a device driver writer can improve performance by setting a bit in the unit control block (UCB) initialization routine to indicate that I/O post processing should be done on the local CPU. This bit is in the UCB\$L_STS field and is called UCB\$[V,M]_IOPOST_LOCAL. It causes IOC\$REQCOM and IOC\$ALTREQCOM to queue the IRP to the CPU-specific post processing queue and do a SOFTINT #IPL\$_IOPOST, if necessary.

Note

Because of ordering requirements, network drivers should not utilize the IOPOST_LOCAL feature.

4.9 Fast Path Performance Enhancements

Prior to OpenVMS Alpha Version 7.3–1, all hardware interrupts took place on the primary CPU. Interrupts from Fast Path enabled devices would have to be redirected from the primary CPU to a “preferred” CPU. This redirection still involved the primary CPU, and also incurred interprocessor overhead. Furthermore, the algorithm for assigning preferred CPUs to ports did not take into account the performance implications of Resource Affinity Domains (RADs).

4.9.1 Distributed Interrupts

Starting with OpenVMS Alpha Version 7.3–1, hardware interrupts for a Fast Path port can be directed to an interrupt CPU, which need not be the primary CPU. Fast Path performance is greatly enhanced when the preferred CPU for a Fast Path port is also the interrupt CPU for that port. In this case, interrupts go directly to the interrupt CPU, eliminating all I/O processing on the primary CPU. This major Fast Path performance enhancement is known as distributed interrupts.

You can use SHOW DEVICES/FULL to display the interrupt CPU ID of a Fast Path port. (If a system does not support distributed interrupts, the interrupt CPU is the primary CPU and does not appear in the SHOW DEVICES display.)

Note

Distributed interrupts are supported on AlphaServer DS20, ES40/45, and GS series systems on Fibre Channel, CI, and some SCSI ports.

System Management Features

4.9 Fast Path Performance Enhancements

4.9.2 Fast Path Code and RAD

The Fast Path code now takes into account the performance implications of the port's Resource Affinity Domain (RAD) in assigning preferred CPUs and interrupt CPUs. Systems without RADs are treated as systems consisting of one RAD. In some systems that support distributed interrupts, not all CPUs can handle them. An interrupt CPU must either be the primary CPU or must be in the RAD of the port.

Although the Fast Path code attempts to take all this into consideration in assigning preferred CPUs, you can use the SET DEVICE/PREFERRED_CPUS command to limit activity to a subset of available CPUs. In a Galaxy configuration, its choice can be overridden by the movement of CPUs in and out of an instance. The default value for PREFERRED_CPUS is NOPREFERRED_CPU, which gives the Fast Path code the greatest latitude in optimizing for performance.

For More Information On	See
Fast Path	<i>VMS I/O User's Reference Volume</i>
RADs	<i>OpenVMS Alpha Partitioning and Galaxy Guide</i>
Fibre Channel, SCSI, and CI configurations	<i>Guidelines for OpenVMS Cluster Configurations</i>

4.10 File Service Extensions

OpenVMS Version 7.2 implemented Extended File Specifications. OpenVMS Version 7.3–1 provides new file system features to further comply with POSIX requirements. These features are supported for ODS-5 volumes, which you can now use as system disks. The result is greater flexibility for OpenVMS Alpha systems to store, manage, and access files created by Windows 95, Windows NT, and UNIX applications. Applications now require less change in porting from these platforms and POSIX compliant environments to OpenVMS Alpha.

The file service extensions consist of support for the following:

- Additional file times
- Access dates
- Hard links
- Case sensitivity

4.10.1 Additional File Times

To support POSIX-compliant file timestamps on ODS-5 disks, OpenVMS Alpha Version 7.3–1 includes three new file attributes:

- ATR\$C_ACCDATE — corresponds to POSIX `st_atime` and reflects the last time a file was accessed.
- ATR\$C_ATTDATE — corresponds to POSIX `st_ctime` and reflects the last time a file attribute was modified.
- ATR\$C_MODDATE — corresponds to POSIX `st_mtime` and reflects the last time data was modified.

Modifications to the file header are recorded as ATTDATE unless the file is actually accessed. The REVDATA ACP/QIO attribute is the most recent of the MODDATE and ATTDATA timestamps. A new ACP/QIO attribute returns the stored REVDATA.

When a file is closed, if “norecord” is set, ACCDATE and REVDATA are not altered. Otherwise, if data has been read from the file, closing a file updates the file’s access date. If data has been written to the file, closing a file updates the file’s modification date.

Because access dates must be written out to disk, using these new file attributes impacts performance. The system manager can use the following command to enable or disable access date support and the frequency for changing access dates:

```
$ SET VOLUME/VOLUME_CHARACTERISTICS=( [[NO]HARDLINKS,] [[NO]ACCESS_DATES[=delta-time]])
```

To limit the performance impact if a file is accessed frequently, you can suppress update of the access time if the change is small. A delta time is used to determine when a new access time is significant.

For an example of how to set access dates using the SET VOLUME/VOLUME_CHARACTERISTICS command, see Section 4.10.2.

4.10.1.1 ACP/QIO Access Dates

Changes in access dates for ACP/QIO depend on whether file dates are being read or maintained.

4.10.1.2 Compaq C Run-Time Library Dates

For information about how to set correct time values, refer to the descriptions of the `utime()`, `stat()`, and `fstat()` functions in the *Compaq C Run-Time Library Reference Manual for OpenVMS Systems*.

4.10.2 DCL Access Dates

To enable automatic update of access dates on ODS-5 volumes, use the SET VOLUME/VOLUME_CHARACTERISTICS command. For example:

```
$ SET VOLUME/VOLUME_CHARACTERISTICS=ACCESS_DATES=[delta-time] NODE$COE1
```

The default value for *delta-time* is 1 second, chosen to comply with the “seconds since EPOCH” time interface required by POSIX `st_atime`. A site can choose a larger delta time to reduce overhead if 1-second granularity is not required.

Another way to enable automatic update of access dates is to use the INITIALIZE/VOLUME_CHARACTERISTICS=ACCESS_DATES command, as follows:

```
$ INITIALIZE/VOLUME_CHARACTERISTICS=ACCESS_DATES NODE$COE1  
$ MOUNT NODE$COE1
```

To disable access date support on a volume, use the SET VOLUME/VOLUME_CHARACTERISTICS=NOACCESS_DATES command. This command affects only the node on which the command is issued. Other nodes are not affected by the change until the next time the volume is mounted.

The DCL commands DIRECTORY or DUMP/HEADER support the new timestamps. Use the following qualifiers:

System Management Features

4.10 File Service Extensions

Qualifier	Description
/DATE=ACCESSED	Specifies the last access date.
/DATE=ATTRIBUTES	Specifies the last attribute modification date.
/DATE=DATA_MODIFIED	Specifies the last data modification date.

For example, the following command displays the last time data was read from the file:

```
$ DIRECTORY/DATE=ACCESSED
```

4.10.3 Hard Links

A link, or directory entry, is an object in a directory that associates a file name and a version number with a specific file. All links on a volume must represent files on the same volume.

With the introduction of hard link support in OpenVMS Alpha Version 7.3–1, OpenVMS now supports three kinds of links on ODS-5 volumes: primary links, aliases, and hard links.

OpenVMS Alpha supports files with zero or more links. The first link to a file is referred to as the *primary link* and is distinguished by having the directory ID and name of the link stored in the file header. Additional links are either *aliases* or *hard links*, depending on whether the volume on which the file resides has hard links enabled or disabled.

On OpenVMS Alpha systems, you can enable hard links on ODS-5 volumes. On volumes where hard links are not enabled, nonprimary links are aliases. If you choose to enable hard links, you cannot create OpenVMS aliases for files on that volume. A volume can support either hard links or aliases, but not both.

The essential difference between hard links and aliases is in the effect of a delete operation. What is usually referred to as deleting a *file* is more precisely deleting a *link* to that file. When a link to a file is deleted, the associated file might also be deleted. Whether or not a file is deleted depends on whether the volume on which the file resides has hard links enabled, and on whether a hard link to that file has been created.

If hard links are enabled, a file is actually deleted when there are no more links to that file. If hard links are not enabled and you have not created an alias for a file, only one link to that file exists: the primary link. If you create an alias for the file and you then delete the alias, the file still exists because the primary link to that file has not been deleted. The alias is just another name in a directory for this link. Deleting the primary link deletes the file and leaves the alias entries.

Attempting to access a file through an alias link to a deleted file results in a “no such file” error. With a primary link and hard links, many links exist. You delete a file when you delete both the primary link and all hard links to that file.

An important related consideration is disk quota. On OpenVMS, the size of each file is charged to the file owner’s disk quota. If other users create hard links to a file, they are not charged disk quota. The file’s owner can delete any links to the file in directories the owner can access, whereas hard links in other users’ directories might cause the file to be retained; the owner continues to be charged for its quota.

When you enable hard link support on an existing volume, be sure to also use the ANALYZE/DISK/REPAIR command to ensure the proper operation of hard links.

System Management Features

4.10 File Service Extensions

OpenVMS supports hard links, or aliases, to directories as well as to files. Most UNIX systems limit hard links to normal files only. The following table shows the commands you use with hard links:

To	Use
Enable hard links	INITIALIZE or SET VOLUME
Disable hard links	SET VOLUME device/VOLUME_CHARACTERISTICS=NOHARDLINKS
Create a hard link (or an alias for a file)	SET FILE/ENTER

Examples

To enable hard links on a mounted Files-11 volume, use the following command:

```
$ SET VOLUME/VOLUME_CHARACTERISTICS=HARDLINKS
```

To initialize an ODS-5 disk with hard links enabled, use the following command:

```
$ INIT/VOLUME_CHARACTERISTICS=HARDLINKS
```

If you have a volume that has hard links enabled, and you want to disable hard links, use the following command:

```
$ SET VOLUME SYS$DISK/VOLUME_CHARACTERISTICS=NOHARDLINKS
```

Note that disabling hard links can result in some strange file behavior. For example, assume that you have disabled hard links, as shown in the preceding example, but you would like to create an alias for your file FOO. First you create an alias for FOO.A called FOO.B, as follows:

```
$ CREATE FOO.A
$ SET FILE FOO.A/ENTER=FOO.B
```

Now delete the original file:

```
$ DELETE FOO.A;1
```

However, if you look for FOO.B on the volume by entering the DIRECTORY command, you receive a “file not found” error because the primary link to that file no longer exists. To fix this problem, or to check the number of hard links to a file, enter the following command:

```
$ ANALYZE/DISK/REPAIR
```

ANALYZE/DISK/REPAIR counts the number of directory entries that reference each file and sets the link count if it is incorrect. Before you create aliases for files on disks that have previously had hard links enabled, be sure to use the ANALYZE/DISK/REPAIR command to set the link count correctly. If you are unsure whether hard links are currently enabled or disabled, use the SHOW DEVICE/FULL command.

To report link counts, use DIRECTORY/FULL and DUMP/HEADER. To check the number of links, enter the following command:

```
$ DIRECTORY/LINK
```

System Management Features

4.10 File Service Extensions

4.10.4 Case-Sensitive File Operations

Traditionally, OpenVMS stored all alphabetic characters in filename specifications as uppercase characters. In addition, file system operations using filename specifications were case insensitive.

The introduction of Extended File Specifications enabled system tools and applications to store and display file specifications containing lowercase as well as uppercase alphabetic characters on ODS-5 volumes.

Filename specification operations were still case insensitive.

With OpenVMS Version 7.3–1 it is now possible for tools and applications to distinguish among filename specifications containing the same alphabetic characters that differ in case only.

You can set processes to ignore or notice the case sensitivity of file names.

Note

Enable case sensitivity only when it is known to be supported by the layered product or application you are working with.

To match all case variants, use the `SET PROCESS/CASE_LOOKUP=BLIND` command. If your process is set to `CASE_LOOKUP=BLIND` and you create more than one file with the same name differing, only in case, DCL treats these files as new versions of the older file and converts them to the same case as the original file.

In the following example, DKA500 is an ODS-5 disk.

```
$ SET DEFAULT DKA500:[TEST]
$ SET PROCESS /CASE=BLIND /PARSE_STYLE=EXTENDED
$ CREATE COEfile.txt
Ctrl/Z
$ CREATE COEFILE.TXT
Ctrl/Z
$ CREATE CoEfile.TXT
Ctrl/Z
$ DIRECTORY
Directory DKA500:[TEST]
COEfile.txt;3
COEfile.txt;2
COEfile.txt;1
```

To create a case-sensitive process environment, use the `SET PROCESS/CASE=SENSITIVE` command. If your process is set to `CASE=SENSITIVE` and you create more than one file with the same name differing only in case, DCL treats subsequent files as new files and lists them as such. In the following example, DKA500 is an ODS-5 disk.

```
$ SET DEFAULT DKA500:[TEST]
$ SET PROCESS /CASE=SENSITIVE /PARSE_STYLE=EXTENDED
$ CREATE COEfile.txt
Ctrl/Z
$ CREATE COEFILE.TXT
Ctrl/Z
$ CREATE CoEFILE.txt
Ctrl/Z
$ DIRECTORY
```

```
Directory DKA500:[TEST]
CoEfile.txt;1
COEFILE.TXT;1
COEfile.txt;1
```

Although an ODS-5 volume preserves the case of a file as it is first entered, file searches are performed in a case-blind manner. Therefore, be careful when you do file comparisons, such as using DCL string functions like .EQS. and F\$LOCATE, in a DCL command procedure.

If you are using an application that expects case sensitivity, or if you depend on case sensitivity in your environment, set your process to /CASE=SENSITIVE. Be aware that using case sensitivity can create problems, if you do not pay attention to your environment.

The default is SET PROCESS /CASE=BLIND /PARSE_STYLE=EXTENDED.

4.10.4.1 Record Management System (RMS) Case Sensitivity

RMS uses the process default for case sensitivity as described in Section 4.10.4. The NAML block was introduced in OpenVMS Alpha Version 7.2 to support long file names. The NAML block has a new field, NAML\$V_CASE_LOOKUP, to override the process default case sensitivity.

Within NAML\$V_CASE_LOOKUP, you can set the following values for case sensitivity:

Field Name	Description
NAML\$C_CASE_LOOKUP_BLIND	Set by the user to tell RMS to ignore case when creating, deleting, and searching for files.
NAML\$C_CASE_LOOKUP_SENSITIVE	Set by the user to tell RMS to include case as a criteria when creating, deleting, and searching for files.

If NAML\$V_CASE_LOOKUP is zero, or if a NAML block is not used, the current process setting is used.

4.10.4.2 ACP/QIO Case Sensitivity

Directory entries are listed in case-blind order. Case-sensitive operations occur when a new FIB option, FIB\$V_CASE_SENSITIVE, is set. ACP/QIO operations with the bit clear are case blind. The process default for case sensitivity does not apply to XQP.

4.10.4.3 System Services That Control the Process Case Sensitivity

The following changes have been made to system services to support case sensitivity:

- New values for properties for the \$SET_PROCESS_PROPERTIES system service
- Changes to the \$GETJPI system service

The \$SET_PROCESS_PROPERTIES system service sets a simple value associated with a service. OpenVMS Alpha Version 7.3-1 supports the following two new property codes for case sensitivity:

System Management Features

4.10 File Service Extensions

Property Code	Description
PPROP\$C_CASE_LOOKUP_TEMP	The type of case lookup to use. This value is set for only the life of the image. The value reverts to the permanent style on image rundown. Valid values are PPROP\$K_CASE_BLIND and PPROP\$K_CASE_SENSITIVE.
PPROP\$C_CASE_LOOKUP_PERM	The type of case lookup to use. This value is set for the life of the process unless the style is set again. The value reverts to the permanent style on image rundown. Valid values are PPROP\$K_CASE_BLIND and PPROP\$K_CASE_SENSITIVE.

The two new item codes for the \$GETJPI system service are:

- JPI\$_CASE_LOOKUP_TEMP
- JPI\$_CASE_LOOKUP_PERM

These item codes return the values that are set by the \$SET_PROCESS_PROPERTIESW system service, which can be either PPROP\$K_CASE_BLIND or PPROP\$K_CASE_SENSITIVE.

For more information about the \$SET_PROCESS_PROPERTIESW system service, refer to the *OpenVMS System Services Reference Manual*.

4.10.4.4 DCL Case Sensitivity

The F\$GETJPI lexical function has two new item codes, CASE_LOOKUP_IMAGE and CASE_LOOKUP_PERM, to return information about the case sensitivity of a process.

To change the case of a file name, use the RENAME command. You must previously have set your process to CASE=SENSITIVE for RENAME to work properly.

For example, to change the case of a file named JANCALENDAR.TXT, enter the following command:

```
$ RENAME JANCALENDAR.TXT JanCalendar.txt
```

This RENAME command renames JANCALENDAR.TXT to JanCalendar.txt.

4.10.4.5 C RTL Support

As of OpenVMS Alpha Version 7.3–1, the Compaq C Run-Time Library has greatly improved support for Extended File Specifications, with 250 of the 254 ODS-5 8-bit characters supported, as opposed to only 214 supported previously. Furthermore, file names can now be reported without file types.

To enable the new support, you must define one or more C RTL logical names. The settings of these logical names affect the behavior of C applications at run-time.

- DECC\$EFS_CHARSET

With DECC\$EFS_CHARSET enabled, UNIX names may contain ODS-5 extended characters. Support includes multiple dots and all ASCII characters in the range 0-255 except <NUL>, (/), ("), (*), and (?).

Unless DECC\$FILENAME_UNIX_ONLY is enabled, some characters may be interpreted as OpenVMS characters, depending on the context. These are (:), (l), (<), (^), (;).

System Management Features

4.10 File Service Extensions

- **DECC\$DISABLE_TO_VMS_LOGNAME_TRANSLATION**
With **DECC\$DISABLE_TO_VMS_LOGNAME_TRANSLATION** enabled, the conversion routine `decc$toVMS()` treats the first element of a UNIX style name as a logical name only if there is a leading slash (/).
- **DECC\$FILENAME_UNIX_NO_VERSION**
With **DECC\$FILENAME_UNIX_NO_VERSION** enabled, OpenVMS generation numbers are not supported in UNIX style file names.
With **DECC\$FILENAME_UNIX_NO_VERSION** disabled, in UNIX style names, generation numbers are reported preceded by a dot.
- **DECC\$FILENAME_UNIX_REPORT**
With **DECC\$FILENAME_UNIX_REPORT** enabled, all file names are reported in UNIX style unless the caller specifically selects OpenVMS style. This applies to `getpwnam()`, `getpwuid()`, `argv[0]`, `getname()`, and `fgetname()`.
With **DECC\$FILENAME_UNIX_REPORT** disabled, and unless it is specified in the function call, file names are reported in OpenVMS style.
- **DECC\$READDIR_DROPDOTNOTYPE**
When **DECC\$READDIR_DROPDOTNOTYPE** is enabled and files are being reported in UNIX style, `readdir()` reports the trailing dot for files with no file type only if the file name contains a dot.
With **DECC\$READDIR_DROPDOTNOTYPE** disabled, all files without a file type are reported with a trailing dot.
- **DECC\$RENAME_NO_INHERIT**
With **DECC\$RENAME_NO_INHERIT** enabled, the new name for the file does not inherit anything from the old name. The new name must be specified completely.
With **DECC\$RENAME_NO_INHERIT** disabled, the new name inherits missing components of the file name such as the device, directory, file type and version from the old file in the same way as the DCL **RENAME** command.

Enter the command **DEFINE name ENABLE** to define each logical name you need. For example:

```
$ DEFINE DECC$EFS_CHARSET ENABLE
```

To disable this logical name, enter:

```
$ DEFINE DECC$EFS_CHARSET DISABLE
```

There are six unsupported ODS-5 characters.

- The OpenVMS file system does not support two characters that are not restricted on UNIX, as follows:

042	0x2a	(*)	Asterisk
063	0x3f	(?)	Question mark
- The C RTL and UNIX cannot handle the following two characters, which are supported by the OpenVMS file system:

000	0x00	(\0)	NUL (the string terminating character)
047	0x2f	(/)	Slash (the directory delimiter)

System Management Features

4.10 File Service Extensions

- The C RTL restricts two other characters, as follows:

033 0x21 (!)	Exclamation point (interpreted as a node delimiter when preceded by characters acceptable in a node name)
034 0x22 (")	Double quotation marks (reserved by the C RTL for use as a delimiter for quoting OpenVMS characters or elements of file names with no UNIX equivalent; quotation marks are currently used similarly in DECnet file specifications)

4.11 New Compaq Graphical Configuration Manager (GCM)

It is now easier for system managers to administer partitioned OpenVMS systems. The Compaq Graphical Configuration Manager (GCM) for OpenVMS is a portable client/server application that provides a visual means of viewing and controlling the configuration of partitioned AlphaServer systems running OpenVMS.

A GCM server runs on each OpenVMS partition in one or more AlphaServer systems.

The GCM client is a JAVA-based application that can run on any operating system that supports the JAVA runtime environment (JDK V1.2.2 or higher) and a TCP/IP network. From a GCM client, an OpenVMS system manager can establish a secure connection to one or more GCM servers, and perform the following functions:

- Display the configuration of partitioned AlphaServer systems
- Utilize hot-swap characteristics of the current hardware platform
- Execute distributed commands among OpenVMS partitions
- Reassign resources among OpenVMS soft partitions
- View resource-specific characteristics
- Shutdown or reboot one or more partitions
- Invoke additional management tools
- Create and engage Galaxy configuration models
- View online documentation

For more detailed information, refer to the *OpenVMS Alpha Partitioning and Galaxy Guide*.

4.12 Kerberos for OpenVMS

Starting with OpenVMS Alpha Version 7.3–1, Kerberos for OpenVMS ships as part of the OpenVMS operating system. Previously, Kerberos shipped as a layered product.

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography.

For more information, refer to the *Kerberos Version 1.0 Security Client for OpenVMS Release Notes*.

4.13 New LANCP Qualifiers for SET DEVICE and SHOW DEVICES

Table 4–1 lists the new qualifiers for the SET DEVICE command.

Table 4–1 SET DEVICE Command Qualifiers

Qualifier	Meaning
/AUTONEGOTIATE /NOAUTONEGOTIAGE	Enables or disables the use of auto-negotiation to determine the settings. You need to disable link auto-negotiation only when connected to a switch or device that does not support auto-negotiation. /AUTONEGOTIAGE is the default.
/CONTENDER /NOCONTENDER	Specifies that the Token Ring is to participate in the Monitor Contention process when it joins the ring. The /NOCONTENDER qualifier directs the device not to challenge the current ring server.
/DEVICE_ SPECIFIC=FUNCTION	Allows some device-specific parameters to be adjusted. For the device-specific function commands, see the <i>OpenVMS System Management Utilities Reference Manual: A–L</i> .
/JUMBO /NOJUMBO	Enables the use of jumbo frames on a LAN device. Only the Gigabit Ethernet NICs support jumbo frames. /NOJUMBO is the default.

Table 4–2 lists the new qualifiers for the SHOW DEVICES command.

Table 4–2 SHOW DEVICES Command Qualifiers

Qualifier	Meaning
/INTERNAL_COUNTERS	Displays internal counters. By default, it does not display zero counters. To see all counters, including zero, use the additional qualifier /ZERO. To see debug counters, use the additional qualifier /DEBUG.
/TRACE	Displays LAN driver trace data.

For more information, see the *OpenVMS System Management Utilities Reference Manual: A–L*.

4.14 LIBDECOMP.COM Enhancements

Several enhancements have been made to the Library Decompression utility (LIBDECOMP.COM):

- In previous releases, LIBDECOMP.COM could perform only one function: expand (decompress) library files. Now the command procedure accepts parameters, which allow you to select any of the following four functions:
 - EXPAND (default) — Expands (decompresses) libraries that ship in data-reduced (compressed) format.
 - REDUCE — Reduces (compresses) libraries to the data-reduced format in which they originally shipped.
 - LIST — Lists all the VAX and Alpha libraries known to the utility. The list displays the size and status (reduced or expanded) of libraries on your system.

System Management Features

4.14 LIBDECOMP.COM Enhancements

- HELP — Outputs a brief help display that outlines the parameters and rules for their use.
- The selection of libraries that ship in data-reduced format has been changed to maximize the benefits of this utility.

Table 4–3 lists all the libraries known to LIBDECOMP.COM on both VAX and Alpha systems.

Table 4–3 Libraries Recognized by LIBDECOMP.COM

Library Name	Platform	Description
[SYSHLP] directory; help library files (.HLB)		
ACLEDT.HLB	Both	Access Control List Editor help
BKM\$HELP.HLB	Both	Backup Manager help
DBG\$HELP.HLB	Both	OpenVMS Debugger help
DBG\$UIHELP.HLB	Both	OpenVMS Debugger help
EDTHELP.HLB	Both	EDT editor help
EVE\$HELP.HLB	Both	EVE editor help
EVE\$KEYHELP.HLB	Both	EVE keypad help
EXCHNGHLP.HLB	Both	Exchange utility help
HELPLIB.HLB	Both	DCL help
LANCP\$HELP.HLB	Both	LAN Control Program help
LATCP\$HELP.HLB	Both	LAT Control Program help
MAILHELP.HLB	Both	Mail utility help
NCPHELP.HLB	Both	Network Control Program help
SDA.HLB	Both	System Dump Analyzer help
SHWCLHELP.HLB	Both	Show Cluster utility help
SYSGEN.HLB	Both	System Generation utility help
SYSMANHELP.HLB	Both	System Management utility help
TPUHELP.HLB	Both	Text Processing Utility help
UAFHELP.HLB	Both	Authorize utility help
[SYSLIB] directory; macro library files (.MLB)		
LANIDDEF.MLB	Alpha only	LAN internal driver macros
LIB.MLB	Both	Operating system macros
STARLET.MLB	Both	Operating system macros
SYSDLMLB.MLB	VAX only	System build files

(continued on next page)

System Management Features 4.14 LIBDECOMP.COM Enhancements

Table 4–3 (Cont.) Libraries Recognized by LIBDECOMP.COM

Library Name	Platform	Description
[SYSLIB] directory; object library files (.OLB)		
DECCRTL.OLB	VAX only	Compaq C Run-time Library
STARLET.OLB	Both	System object library and run-time library
SYSBLDLIB.OLB	VAX only	System build files
VAXCRTL.OLB	Both	Compaq C RTL routine name entry points; VAX G_floating double-precision, floating-point entry points
VAXCRTLD.OLB	Alpha only	Limited support for VAX D_floating double-precision, floating-point entry points
VAXCRTLDX.OLB	Alpha only	VAX D_floating support; support for /L_DOUBLE_SIZE=128 compiler qualifier
VAXCRTLT.OLB	Alpha only	IEEE T_floating double-precision, floating-point entry points
VAXCRTLTX.OLB	Alpha only	IEEE T_floating support; support for /L_DOUBLE_SIZE=128 compiler qualifier
VAXCRTLX.OLB	Alpha only	G_floating support; support for /L_DOUBLE_SIZE=128 compiler qualifier
VMS\$VOLATILE_PRIVATE_INTERFACES.OLB	Alpha only	OpenVMS bugcheck processing codes
[SYSLIB] directory; text library files (.TLB)		
ERFLIB.TLB	Both	ANALYZE/ERROR device descriptions
LIB_ADA_SUBSET.TLB	Both	Ada programmers toolkit of operating system definitions
NTA.TLB	Both	Files to build against NTA facility
STARLET_RECENT_ADA_SUBSET.TLB	Both	Ada programmers toolkit of operating system definitions
STARLETSD.TLB	Both	STARLET definitions used during layered product installations
SYS\$LIB_C.TLB	Alpha only	Header files for C language; derived from LIB
SYS\$STARLET_C.TLB	Both	Public header files for Compaq C

The following libraries used to be compressed but are no longer shipped in data-reduced format:

```
[SYSHLP]FMG_HELP_FILE_AMERICAN_ENGLISH.HLP
[SYSHLP]ANALAUDIT$HELP.HLB
[SYSHLP]ANLRMSHLP.HLB
[SYSHLP]DISKQUOTA.HLB
[SYSHLP]EDFHLP.HLB
[SYSHLP]INSTALHLP.HLB
[SYSHLP]MNRHELP.HLB
[SYSHLP]PATCHHELP.HLB
[SYSHLP]PE$HELP.HLB
[SYSHLP]PHONEHELP.HLB
[SYSHLP]TECO.HLB
[SYSHLP]TFF$TFUHELP.HLB
[SYSLIB]DECCCURSE.OLB
[SYSLIB]DECCRTL.G.OLB
```

System Management Features

4.14 LIBDECOMP.COM Enhancements

```
[SYSLIB]IMAGELIB.OLB  
[SYSLIB]VAXCCURSE.OLB  
[SYSLIB]VAXCRTL.G.OLB
```

The following libraries were not compressed in the past but now ship in data-reduced format:

```
[SYSHLP]BKM$HELP.HLB  
[SYSHLP]LANCP$HELP.HLB  
[SYSLIB]LANIDDEF.MLB  
[SYSLIB]LIB_ADA_SUBSET.TLB  
[SYSLIB]NTA.TLB  
[SYSLIB]STARLET_RECENT_ADA_SUBSET.TLB  
[SYSLIB]SYSBLDLIB.OLB  
[SYSLIB]SYSBLDMLB.MLB  
[SYSLIB]VAXCRTLDX.OLB  
[SYSLIB]VAXCRTLT.X.OLB  
[SYSLIB]VAXCRTLX.OLB
```

For detailed information about using LIBDECOMP.COM and its parameters, refer to the *OpenVMS System Manager's Manual*.

4.15 ODS-5 Volume Structure

ODS-5 is an optional volume structure that provides support for longer file names with a greater range of legal characters. The legal character set has been extended. Now you can use all the characters from the ISO Latin-1 Multinational character set, *except* the asterisk (*) and the question mark (?), in file names. Some of the characters from the ISO Latin-1 character set require an escape character to be interpreted properly. For more information, refer to the *OpenVMS User's Manual*.

In previous versions of OpenVMS, ODS-5 volumes could not be used as system disks, and it was recommended that ODS-5 disks be used in homogeneous Alpha clusters only. These restrictions have been removed. OpenVMS Version 7.3-1 supports the use of ODS-5 volumes as system disks and in heterogeneous Alpha clusters.

The ODS-5 volume structure supports many new file system features, including hard links and support for case sensitivity. For detailed information about the new file system features, see Section 4.10. For more information about the system management of ODS-5 disks, refer to the *OpenVMS System Manager's Manual, Volume 1: Essentials*.

4.16 OpenVMS Cluster New Features

This section briefly describes the new OpenVMS Cluster features introduced in this release.

4.16.1 OpenVMS Cluster Performance Improvements

Many modules that are used to send messages or transfer data in an OpenVMS Cluster system have been optimized for speed of execution on the Alpha architecture. Several major modules have been completely rewritten, routine by routine, for performance, correctness, and clarity.

The resultant images are significantly smaller, invoke far fewer memory accesses, and generate much more efficient code.

System Management Features

4.16 OpenVMS Cluster New Features

The benefits of these changes are:

- Reduced system overhead for frequent cluster operations
- Improved throughput for cluster locking, messages, and data transfers
- Improved throughput for local locking
- Reduced hold time for the IOLOCK8 spinlock
- Introduction of SYS\$CLUSTER_MON.EXE image for automatic additional sanity checks when SYSTEM_CHECKS is enabled

Background

The cluster communication subsystem consists of several layers:

- System application (SYSAP)
- System Communications Services (SCS)
- Port (SCS port driver, such as PEDRIVER or PNDRIVER)
- Device (for example, LAN adapter or CI adapter)

The major system applications used by OpenVMS are the disk class driver (DUDRIVER), the MSCP server (MSCP), and the cluster connection manager (part of the SYS\$CLUSTER image).

SYS\$CLUSTER is a shared system application encompassing many subfacilities, notably:

- Distributed lock manager
- Cluster server process (CSP) messages
- Clusterwide logical names messages
- Clusterwide process services (CWPS)
- Distributed Get Lock Information Service (GETLKI)
- Cluster connection manager (CNXMAN)
- Distributed write bitmap messages

These subfacilities have many subfunctions and services themselves. For example, the distributed lock manager has services for sending \$ENQ and \$DEQ messages, remastering resource trees, deadlock detection, failover recovery, and others.

To send a message:

1. A system application invokes common interfaces to SCS routines that handle messages or data transfer.
2. The SCS layer calls through a common interface to access port-specific routines for the adapters used to reach the remote system or controller.
3. The Port layer calls the specific device driver or adapter interface.

Incoming messages reverse this ordering. Within the connection manager, an additional layer to provide sequenced message and block transfer services is available to all of its subfacilities.

System Management Features

4.16 OpenVMS Cluster New Features

Revised Components

The following subsystems have been totally or incrementally rewritten for speed of execution on Alpha, resulting in performance improvement for all system applications in the cluster:

- Entire mainline code for the SCS layer
- Entire SCS Fast Path layer (DUdriver is the only client)
- SCS port interface layer for:
 - PEdriver (LAN)
 - PNdriver (CI)
 - PMdriver (Shared Memory CI for Galaxy)
 - Remaining port types
- CI interrupt handler
- Common I/O interrupt dispatcher
- Common subroutines within the local lock manager
- Various common system macros

The following subsections of SYS\$CLUSTER have also been totally rewritten to optimize performance:

- Connection manager (common across subcomponents):
 - Sequenced message send/receive handler (completion of updates begun in OpenVMS Version 7.3)
 - Block transfer handling
 - Resource allocation/deallocation/initialization
- Distributed lock manager:
 - ALL send/receive LCKMGR message paths
 - Streamlined mainline of block transfer remastering
- Write bitmap:
 - Received message dispatching

4.16.2 Fibre Channel Driver Optimizations

For OpenVMS Alpha Version 7.3–1, the Fibre Channel driver has been optimized to reduce I/O lock hold time by 3–6 microseconds per I/O, resulting in significant I/O performance improvements. This optimization, coupled with the coalescing of the I/O completion interrupts, has reduced I/O lock times by as much as 50%, potentially doubling Fibre Channel throughput.

Unlike the coalescing of the I/O completion interrupts, the Fibre Channel driver optimizations cannot be backported to earlier versions of OpenVMS Alpha.

For more information about Fibre Channel configurations, refer to *Guidelines for OpenVMS Cluster Configurations*.

4.16.3 I/O Interrupt Coalescing for Fibre Channel Configurations

OpenVMS Alpha Version 7.3–1 contains a major enhancement to the processing of I/O completion interrupts in the host bus adapter. Instead of being sent one at a time, the I/O completion interrupts are grouped together and then transmitted as a group. This major enhancement improves I/O performance in environments with high I/O work loads. Initial tests show a 25% reduction in IOLOCK8 hold time, which translates directly into a 25% increase in I/O throughput performance.

This feature will be backported to OpenVMS Alpha Version 7.2–2 and OpenVMS Alpha Version 7.3.

For more information about Fibre Channel configurations, refer to *Guidelines for OpenVMS Cluster Configurations*.

4.16.4 MSCP Served Devices: Increase from 512 to 1000

OpenVMS Alpha Version 7.3–1 contains a major enhancement to MSCP serving. The previous limit of 512 disks that could be MSCP served by any system in an OpenVMS Cluster system has been raised to 1000. This allows greater flexibility in configuring OpenVMS Cluster storage. A further increase making the supported limit greater than 1000 will be considered for a future release.

Note that this increase affects Alpha systems, only. There is no intent to increase the number of disks served by VAX nodes. A VAX node can still be a client of as many disks as are presented in the cluster, within memory resource limits.

For more information about MSCP serving, refer to *OpenVMS Cluster Systems* and *Guidelines for OpenVMS Cluster Configurations*.

4.16.5 Multipath Failover to an MSCP-Served Path in SCSI and Fibre Channel Configurations

In a multipath SCSI or Fibre Channel configuration, OpenVMS supports failover, from one path to another path, to a device. Establishing multiple paths to a device has the following advantages:

- Increases the availability of the device
- Provides higher overall performance

Prior versions of OpenVMS Alpha supported failover between direct paths.

OpenVMS Alpha Version 7.3–1 introduces support for failover between direct-attached and MSCP-served paths to disks in the following OpenVMS Cluster configurations:

- Multihost SCSI
- Multihost Fibre Channel

If all direct paths to a device are broken, the I/O automatically fails over to an MSCP-served path. When the direct paths are restored, the I/O is automatically rerouted from the MSCP-served paths to the direct-attached paths.

Use the MPDEV_REMOTE system parameter to enable this capability. For more information about multipath support, including failover to an MSCP-served path, refer to *Guidelines for OpenVMS Cluster Configurations*.

System Management Features

4.16 OpenVMS Cluster New Features

4.16.6 Automatic Multipath Balancing for Fibre Channel and SCSI Devices

A major enhancement to multipath failover in OpenVMS Alpha Version 7.3–1 for Fibre Channel and SCSI disk and tape devices is automatic path balancing. The path selection of each device is now biased toward the connected path with the fewest devices using it as a current path.

In addition to the introduction of automatic, multipath path balancing, the path selection algorithm for multipath failover has been modified to improve performance. For more information, see Section 6.7.8, Path Selection by OpenVMS, in *Guidelines for OpenVMS Cluster Configurations*.

4.16.7 Multipath Tape Support, Including Failover, for Fibre Channel Configurations

Tapes are supported in Fibre Channel configurations as of OpenVMS Alpha Version 7.3. You can attach SCSI tape devices to the Fibre Channel via a Fibre-to-SCSI bridge known as the Modular Data Router (MDR). The MDR is dual ported, allowing two paths into the MDR. For example, if an Alpha system has four KGPSA adapters, there are four distinct paths to a tape drive on the Fibre Channel. An Alpha system with four KGPSA adapters leading to a dual-ported MDR actually has eight different paths to a tape drive.

OpenVMS Alpha Version 7.3 did not take advantage of multiple paths. It used only the first path detected during autoconfiguration. The remaining paths were never recognized or made available, even if the first path broke. This single-path model had two limitations:

- One KGPSA must handle all the tape activity.
- There is no failover capability in the event of a path failure.

The first limitation has a workaround that uses the selective storage presentation (SSP) feature of the MDR utility; the second limitation has no workaround at all.

OpenVMS Alpha Version 7.3–1 removes both limitations. All possible paths from an Alpha system to a Fibre Channel tape are configured and made available. You can specify a particular path with the DCL command SET DEVICE/SWITCH. In the event of a broken connection, automatic failover takes place.

Note

Multipath failover is supported between tape devices directly connected to the Fibre Channel that are members of a multipath set. If one member of the set fails, another member provides the tape device to the client.

However, multipath failover between direct and MSCP-served paths is not supported for tape devices.

For more information about tape support in Fibre Channel configurations, refer to *Guidelines for OpenVMS Cluster Configurations*.

4.16.8 New Cluster SCA Circuit and Port Functionality

The following sections explain new functionality available for SCA circuits and ports and new support for SCS dynamic load class. In addition, you can independently enable or disable checksumming on individual PEdriver virtual circuits (VCs). These changes provide a more dynamic connection environment for improving overall performance in an OpenVMS Cluster system.

4.16.8.1 Ability to Set Port and Circuit Priorities

OpenVMS Cluster code has always attempted to assign SCS connections automatically to a circuit on the highest performance cluster interconnect. These assignments were based on the interconnect-specific load class value. Automatic circuit selection does not always produce connection assignments that are optimal for a customer's particular performance or availability requirements. Unfortunately, no mechanism has existed to override automatic circuit selection. This problem has been corrected in OpenVMS Version 7.3-1.

Beginning with OpenVMS Version 7.3-1, you can exercise direct management control over the circuits chosen for SCS connections. To tune your cluster to your specific performance or availability requirements, now you can override automatic selection of the circuit with the highest load class value.

To override automatic circuit selection, use either the SCACP utility or the Availability Manager to assign a management priority value to a specific circuit or SCA port. (A circuit's current priority value is the sum of the local port's management-assigned priority and the management priority value assigned to that circuit.)

Connections are assigned to a circuit with the highest priority. If multiple circuits have the highest priority value, then the circuit with the highest load class is selected. A change in a circuit's current priority has one of the following effects:

- If a circuit's new priority value is higher than another circuit's current priority, the connection moves to the circuit with the higher current priority.
- If a circuit's new priority value is lower than another circuit's current priority, the connection moves to the circuit with the highest current priority.

Circuit Management Priority is Volatile

Whenever a circuit is closed, its management priority setting is lost. This is because the data structure containing information about a circuit is deallocated each time a circuit is closed. When a circuit is reopened, the structure is initialized with default values. Thus, circuit management priority does not propagate across VC closures.

4.16.8.2 Ability to Enable/Disable PEdriver Checksumming

Prior to OpenVMS Version 7.3-1, checksumming could be enabled or disabled on all of the circuits of a node only. This sometimes resulted in unnecessary use of the CPU for circuits on which the additional robustness of checksumming was not needed. Beginning with OpenVMS Version 7.3-1, you can enable or disable checksumming on individual LAN circuits. This lets you specify the use of checksumming for only those circuits where it is needed, such as circuits between the sites of a disaster-tolerant cluster.

System Management Features

4.16 OpenVMS Cluster New Features

4.16.8.3 New SCS Dynamic Load Class Support

Prior to OpenVMS Version 7.3-1, the load class of SCS circuits was determined only by the port's hard-coded load class value. As a result, CI or DSSI circuits were chosen over a GigaBit Ethernet circuit. Beginning with OpenVMS Version 7.3-1, PEdriver dynamically updates the load class value of its SCS circuits to reflect the performance of the underlying LAN path or paths currently in use.

If the circuits have the same priority, a change in a circuit's load class has one of the following effects:

- If a circuit's new load class value is higher than another circuit's current load class, the connection moves to the higher load class circuit.
- If a circuit's new load class value is lower than another circuit's load class, the connection moves to the circuit with the highest load class.

4.17 OpenVMS Registry New Features

New features for the OpenVMS Registry include two enhancements to the Registry Control Program (REG\$CP) and performance improvement through indexing.

4.17.1 Database Version Support

The Registry Control Program, REG\$CP, has been enhanced to support the creation of distinct Registry database versions. The CREATE DATABASE command now accepts a /VERSION qualifier. For OpenVMS Version 7.3-1, a Version 1 or Version 2 database may be specified. By default, CREATE DATABASE creates a Version 2 database.

4.17.2 Value Type Support

In addition to value types of SZ, EXPAND_SZ, MULTI_SZ, and DWORD, the Registry Control Program, REG\$CP, has been enhanced to support the BINARY value type when creating or modifying values (/TYPE_CODE=BINARY). When listing values with BINARY value type, the data is displayed in hexadecimal radix.

Creating and modifying values has also been enhanced with the addition of the /INPUT qualifier to allow the value data to be read from a file. The input data type is specified with embedded keywords in the file records. These keywords are compatible with those found in an exported database file.

The storage data type, as specified by the /TYPE_CODE qualifier, is independent of the input data type. For example, you may specify the value type as SZ (Unicode string) but input the value data in hexadecimal.

Combined, these features allow large amounts of raw data to be entered via REG\$CP. By contrast, entering the data with the /DATA qualifier and the DWORD data type, while allowing raw data to be input, imposes restrictions on the length of the data. Similarly, entering SZ value types with /DATA, while not imposing a length restriction, does impose a restriction on the type of data that can be input, that is, you can enter only Unicode strings. The command line also restricts the amount of data that can be input with /DATA. The /INPUT qualifier provides a way to overcome these restrictions.

4.17.3 New Registry Database Version

The performance of the OpenVMS NT Registry has been improved in Version 7.3-1. This was accomplished primarily by indexing the Registry database. This, however, resulted in the need to create a new Registry database version.

The Registry server will continue to support the previous database format, Version 1.0; however, to take advantage of the indexing, you must convert the current database to Version 2.0.

Converting a Version 1 database to Version 2 is optional. Do not convert the Registry database if you plan to run Registry servers on nodes in a mixed-version cluster, while upgrading your cluster. Registry servers on nodes running versions of OpenVMS prior to Version 7.3-1 would not be able to access a Version 2 database.

For further information on converting an OpenVMS Registry database, refer to the *COM, Registry, and Events for OpenVMS Developer's Guide*.

4.18 SHOW CLUSTER Utility Adds New Fields

The OpenVMS Show Cluster utility (SHOW CLUSTER) is used to monitor nodes in an OpenVMS Cluster and to display information about cluster-specific activity and performance. It has been enhanced to provide additional information about virtual circuits and local ports.

Table 4-4 describes the new fields that you can use with the CIRCUIITS class to display virtual circuit information.

Table 4-4 CIRCUIITS Class New Fields

Field name	Description
LD_CLASS	The circuit's current capacity rating.
MGT_PRIORITY	Priority value assigned to the circuit by management action.
PRIORITY	Circuit's current priority, which is the sum of the management priorities assigned to the circuit and the associated local port.

Table 4-5 describes the new fields that you can use with the LOCAL_PORTS class to display additional local port information.

Table 4-5 LOCAL_PORTS Class New Fields

Field name	Description
LP_LD_CLASS	Hard-coded capacity value of the port, based on the megabits per second rate of the interconnect of the port.
LP_PRIORITY	Management priority assigned to the port.

For more information about the SHOW CLUSTER utility, refer to the *OpenVMS System Management Utilities Reference Manual*.

System Management Features

4.19 New Method for Naming Spawned Subprocesses

4.19 New Method for Naming Spawned Subprocesses

The way OpenVMS names spawned subprocesses has been changed to improve performance.

DCL_CTLFLAGS is a bitmask used to alter default behavior for certain commands on a systemwide basis. At present, only the low bit of the bitmask is defined. The low bit controls the default process-name assignment for a subprocess created using the SPAWN command or LIB\$SPAWN routine.

Prior to OpenVMS Version 7.3-1, if no process name was supplied, the system constructed a name by appending *_n* to the username, where *n* was the next available non-duplicate integer for any process currently in the system. For example, the first spawned process from user SYSTEM would be called SYSTEM_1, the second, SYSTEM_2, and so on. The next available number was chosen, as soon as a gap was found.

A problem with this technique is that determining the next available number is very expensive in terms of performance, because the mechanism attempts to create the process by incrementing names until one is found that is unique. When several subprocesses already exist, the cost of creating the subprocess iteratively becomes even more expensive. When many processes are in the same OpenVMS group, the cost multiplies because process names must be unique throughout the group.

Beginning with OpenVMS Version 7.3-1, the default-constructed process name for subprocesses has changed. Instead of incrementally searching for the next unique number, a random number is chosen to append to the username. Therefore, the first processes that are spawned from user SYSTEM might be SYSTEM_154, SYSTEM_42, SYSTEM_87, and so on. This procedure results in a very high probability of finding a unique name on the first try, because it is unlikely that the same number is already in use. This greatly reduces the cost of process creation, and applications that rely on spawned subprocesses might see a dramatic performance improvement due to this change.

However, some applications might rely on the prior method of assigning subprocess names. The DCL_CTLFLAGS parameter is available to allow you to configure the system as necessary.

Bit 0 of DCL_CTLFLAGS selects the behavior for assigning default subprocess names:

- If the bit is clear, the new behavior is used. If you do not specify a process name, the system assigns the username with a random number suffix. This is the default setting.
- If the bit is set, the prior behavior is used. If you do not specify a process name, the system assigns the username with the next available number.

4.20 SYSMAN Utility Has New Commands and Qualifiers

The SYSMAN utility has new qualifiers for the existing RESERVED_MEMORY ADD and RESERVED_MEMORY MODIFY commands and two new commands, described in Table 4-7.

System Management Features

4.20 SYSMAN Utility Has New Commands and Qualifiers

Table 4–6 New SYSMAN Command Qualifiers

Command	Qualifier	Description
RESERVED_MEMORY ADD	/RAD	Specifies the RAD for the reservation you want to make in the Reserved Memory Registry data file.
RESERVED_MEMORY MODIFY	/RAD	Modifies an existing RAD in the Reserved Memory Registry data file.
	/NEW_RAD	Specifies the RAD you want to substitute in the Reserved Memory Registry data file.

Table 4–7 describes the new SYSMAN commands in OpenVMS Version 7.3–1.

Table 4–7 New SYSMAN Utility Commands

Command	Description
RESERVED_MEMORY EXTEND	<p>The RESERVED_MEMORY ADD command reserves an amount of physical memory, referred to as a memory reservation. Use the RESERVED_MEMORY EXTEND command when you want to add memory sections to accommodate more than one resource affinity domain (RAD) for a single memory reservation. It has two qualifiers:</p> <ul style="list-style-type: none"> • /RAD Specifies one or more RADs for a memory section. • /NORAD Adds a memory section without specifying a RAD.
RESERVED_MEMORY LIST	Previews a reservation as it is currently stored in the Reserved Memory Registry data file. If you don't specify a reservation, all current reservations are displayed.

4.21 New System Parameters

The following sections describe system parameters that are new in OpenVMS Version 7.3–1.

4.21.1 DCL_CTLFLAGS

The new system parameter DCL_CTLFLAGS allows users to continue to use the previous default behavior for assigning subprocess names on a systemwide basis when processes are spawned. Refer to Section 4.19 for more information.

4.21.2 DELPRC_EXIT

Use DELPRC_EXIT to control \$DELPRC system service options that call exit handlers prior to final cleanup and deletion of a process. The following table describes these options:

Option	Description
0	Disable the exit handler functionality with \$DELPRC.
4	Execute kernel-mode exit handlers.
5 (default)	Execute exec- and more privileged-mode exit handlers.
6	Execute supervisor- and more privileged-mode exit handlers.

System Management Features

4.21 New System Parameters

Option	Description
7	Execute user- and more privileged-mode exit handlers.

DELPRC_EXIT is a dynamic parameter.

4.21.3 MPDEV_AFB_INTVL

MPDEV_AFB_INTVL specifies the automatic failback interval in seconds. The automatic failback interval is the minimum number of seconds that must elapse before the system attempts another failback from an MSCP path to a direct path on the same device.

MPDEV_POLLER must be ON to enable automatic failback. You can disable automatic failback without disabling the poller by setting MPDEV_AFB_INTVL to 0. The default is 300 seconds.

4.21.4 RMS_SEQFILE_WBH

RMS_SEQFILE_WBH can enable the RMS write-behind feature as a system default for any unshared sequential disk file if the file is opened for image I/O with write access specified. The following table describes the possible settings:

Setting	Description
0 (default)	Do not enable write-behind feature. Preserve prior behavior of using write-behind only if the user requests it by setting RAB\$V_WBH in RAB\$L_ROP.
1	Enable write-behind feature as system default, including the allocation of at least two local buffers.

RMS_SEQFILE_WBH is a dynamic parameter.

4.22 New Capabilities for SCACP Utility

Beginning with OpenVMS Version 7.3–1, you can use SCACP to manage SCA communications on all OpenVMS Cluster interconnects. These interconnects are in addition to SCA use of LAN devices and LAN paths, which was supported beginning in OpenVMS Version 7.3.

4.22.1 Cluster Ports and Circuits

SCACP allows you to display port and circuit information for any cluster interconnect. You can also set the management priority field in ports or circuits. This lets you optimize the cluster's choice of circuits and ports used for cluster configuration.

4.22.2 New Commands and Qualifiers

Table 4–8 describes the new SCACP SET and SHOW commands in OpenVMS Version 7.3–1 and lists their qualifiers.

System Management Features

4.22 New Capabilities for SCACP Utility

Table 4–8 New SCACP SET and SHOW Commands and Their Qualifiers

Command	Function
SET CIRCUIT	<p>Allows a user to set a management priority value for the selected circuit or circuits. This command can be used with the following qualifiers:</p> <ul style="list-style-type: none"> • /EXCLUDE=(nodename[/PORT=portname[/RSTATION=n]][,...]) Allows you to exclude a specific circuit to a node. If multiple circuits to the same node exist, you can use the /PORT and /RSTATION qualifiers to uniquely identify the circuit. • /PORT=portname[/RSTATION=n] If multiple circuits to the same node exist, you can use the /PORT and /RSTATION qualifiers to uniquely identify the circuit. You can use the /RSTATION qualifier only in conjunction with the /PORT qualifier for the selected circuit or circuits. • /PRIORITY=n Sets the management priority value for the selected circuits. <i>n</i> can be any value between -127 and +127.
SET PORT	<p>Allows a user to set a management priority value for the selected port or ports. This command can be used with the following qualifiers:</p> <ul style="list-style-type: none"> • /EXCLUDE=(portname[,...]) Excludes specific ports from the operation. • /PRIORITY=n Sets the management priority value for the selected ports. <i>n</i> can be any value between -127 and +127.
SHOW CIRCUIT (nodenames)	<p>Shows information about all circuits between this node and other cluster nodes, including LAN circuit information. This command can be used with the following qualifiers:</p> <ul style="list-style-type: none"> • /EXCLUDE=(nodename[/PORT=portname[/RSTATION=n]][,...]) • /EXCLUDE Allows you to exclude a specific circuit to a node. If multiple circuits to the same node exist, you can use the /PORT and /RSTATION qualifiers to uniquely identify the circuit. • /PORT=portname[/RSTATION=n] If multiple circuits to the same node exist, you can use the /PORT and /RSTATION qualifiers to uniquely identify the circuit. You can use the /RSTATION qualifier only in conjunction with the /PORT qualifier.
SHOW PORT	<p>Displays information about all SCA ports on the node, including the LAN port, PEA0. This command can be used with the following qualifiers:</p> <ul style="list-style-type: none"> • /EXCLUDE=(portname[,...]) Excludes specific port names from the display. You cannot use wildcards to specify port names. • /OUTPUT=filespec Creates the specified file and directs the output of the command to this file.

Table 4–9 describes the new SCACP command qualifiers in OpenVMS Version 7.3–1.

System Management Features

4.22 New Capabilities for SCACP Utility

Table 4–9 New SCACP Command Qualifiers

Command	Qualifier	Description
SET VC	/CHECKSUMMING /NOCHECKSUMMING (default)	Enables or disables checksum verification on the VC to each specified node. You can use this command alone or in combination with the system parameter NISCS_PORT_SERV. (Refer to online help for NISCS_POST_SERV for more information.)
VC CLOSURE SIDE-EFFECT		
When you enable or disable checksumming on a circuit, the circuit closes briefly and then reopens using the specified operating mode. This is necessary to ensure that both ends of the circuit are synchronized.		
SHOW LAN_ DEVICE	/EXCLUDE=(landevicename[,...])	Excludes specific LAN devices, which you can use wildcards to specify.

4.23 Ultra3 SCSI Adapter Supported with Fast Path

OpenVMS Alpha Version 7.3–1 supports the KZPEA, an Ultra3 SCSI host adapter that offers a maximum bus speed of 160 MB/s. PKAdriver is the software component that supports KZPEA. The KZPEA is supported for direct-attached storage only.

OpenVMS Alpha Version 7.3–1 also provides Fast Path support for the KZPEA. Fast Path support is designed for use in a multiprocessor system. Fast Path automatically redirects I/O from the primary CPU to different CPUs. You can change it either programmatically or by DCL commands.

Note

Support for the KZPEA adapter on OpenVMS Alpha Version 7.2–2 and Version 7.3 is available in remedial kits.

Fast Path support is not provided for the KZPEA for OpenVMS Alpha Version 7.2–2 or Version 7.3.

For more information about Fast Path support, refer to the *OpenVMS I/O User's Reference Manual*.

4.24 XA Capability Added to DECdtm

DECdtm is a distributed transaction manager that coordinates the activity of resource managers used by an application. DECdtm services are embedded in the OpenVMS operating system and fully support distributed databases using a two-phase commit protocol. The DECdtm services provide the technology and features for distributed processing, ensuring both transaction and database integrity across multiple Compaq resource managers.

DECdtm V2.0 implements the XA standard interface from the X/Open distributed transaction model. The XA interface provides transaction processing interoperability across OpenVMS Oracle/Rdb, OpenVMS Oracle 8i/9i, RMS Journaling, and ACMS. DECdtm V2.0 can interface with XA compliant

System Management Features

4.24 XA Capability Added to DECdtm

transaction managers, such as BEA Tuxedo and Compaq's Reliable Transaction Router (RTR).

DECdtm with XA provides:

- Distributed transaction coordination between applications and Oracle 9i, Oracle RDB, and RMS databases.
- ACMS access to the Oracle 9i databases, in addition to Oracle RDB and RMS.
- Transaction and data integrity, allowing for greater transaction interoperability.

For More Information On	See
DECdtm system services	<i>OpenVMS System Services Reference Manual</i>
Managing DECdtm and XA transactions	<i>OpenVMS System Manager's Manual</i>
XGCP Utility and DECdtm/XA	<i>OpenVMS System Management Utilities Reference Manual</i>
Programming with DECdtm and XA	<i>OpenVMS Programming Concepts Manual, Volume II</i>

Programming Features

This chapter describes new features relating to application and system programming on this version of the Compaq OpenVMS operating system.

5.1 Taking Advantage of Performance-Enhancing Features of Alpha Processors

Each generation of Alpha processors has different characteristics. Applications built to run on one generation of processors continue to run on newer generations. However, newer generations of processors are usually more capable and perform better than previous generations. By rebuilding applications for newer processors, you can take advantage of their new performance-enhancing capabilities and features.

A white paper tells image-providers—ISVs or product developers—how they can take advantage of some of the performance-enhancing features of newer Alpha processors. The HTML and PDF versions of the paper are at the following web site:

<http://www.openvms.compaq.com/doc/>

5.2 ANALYZE Usage Summary Has New Qualifier

The `/SHADOW_MEMBER` qualifier in the Alpha System Analysis tool `ANALYZE Usage Summary` specifies which member of a shadow set contains the system dump to be analyzed, or allows the user to determine what system dumps have been written to the member of the shadow set. For more information, refer to the *OpenVMS Alpha System Analysis Tools Manual*.

5.3 Compaq C Run-Time Library Enhancements

The following sections describe the Compaq C RTL enhancements included in OpenVMS Version 7.3–1. These enhancements provide improved UNIX compatibility and the flexibility of user-controlled feature selections. For more details, refer to the *Compaq C Run-Time Library Reference Manual for OpenVMS Systems*.

5.3.1 Files Larger Than 2 Gigabytes

Support has been added for compiling applications to use file sizes and offsets that are 2 gigabytes or larger. This is accomplished by allowing file offsets of 64-bit integers. Two new functions have been added:

```
fseeko  
ftello
```

These are identical to `fseek` and `ftell`, but they accept or return values of type `off_t`, which allows for a 64-bit variant of `off_t` to be used.

Programming Features

5.3 Compaq C Run-Time Library Enhancements

Modifications to accommodate a 64-bit file offset have been made to the following existing C RTL functions:

```
fstat
ftruncate
ftw
lseek
mmap
stat
truncate
```

The new 64-bit interfaces can be selected at compile time by defining the `_LARGEFILE` feature macro:

```
$ CC /DEFINE=_LARGEFILE
```

5.3.2 Enabling C RTL Features at Compile/Link Time

The C RTL now provides an extensive list of feature switches that affect an application's behavior. They can be used to enable new behaviors or preserve previous, deprecated behaviors. You can enable or disable these features at run time using DECC\$ logical names. For more information, see Section 4.10.4.5 and refer to the *Compaq C Run-Time Library Reference Manual for OpenVMS Systems*.

A programmable interface is also available to sense or set feature switches within an application. By using this interface in a function referenced by the global symbol LIB\$INITIALIZE, you can enable or disable features before the C RTL is initialized.

In many instances, you can enable new behaviors by using feature switches. This will preserve existing behavior as the default. If an application needs to select a different default behavior, a user can then modify the run-time behavior using a logical name.

For some applications, some feature-switch settings may be necessary. In this case, they can be set in a way that a logical name cannot override the value.

There are four functions provided to do this:

```
int decc$feature_get_index(char *name);
char *decc$feature_get_name(int index);
int decc$feature_get_value(int index, int mode);
int decc$feature_set_value(int index, int mode, int value);
```

5.3.3 Enhanced Support for UNIX Style File Names

The Compaq C RTL can be instructed to accept UNIX style file and directory path names that contain the extended ASCII character set supported by OpenVMS Extended File Specifications (EFS) on ODS-5 devices.

Note

This feature is intended for use by applications ported from UNIX systems and not for native OpenVMS applications.

To enable this feature, define the DECC\$FILENAME_UNIX_ONLY logical name to ENABLE before program execution:

```
$ DEFINE DECC$FILENAME_UNIX_ONLY ENABLE
```

Programming Features

5.3 Compaq C Run-Time Library Enhancements

With this feature enabled, all file and path specifications input to the Compaq C RTL are assumed to be UNIX style and therefore can consist of the full, printable ASCII character set (20-7E hexadecimal), with the following exceptions:

- * (asterisk)
- ? (quotation mark)
- / (forward slash)
- " (double quotation mark)

The Compaq C RTL provides any necessary enclosing quotation marks for special characters when presenting the file specification to the OpenVMS file system. OpenVMS file and path specifications that are output by the Compaq C RTL are converted to UNIX style where appropriate, and any special characters quoted by the OpenVMS file system are unquoted before being returned to the Compaq C RTL caller. In addition, the OpenVMS version number field is stripped from the file name on output.

For example, assume the following UNIX style path name is input:

```
./abc.d/data.dat.dat
```

The Compaq C RTL converts it to the following before passing it on to the OpenVMS file system:

```
[.abc^.d]data^.dat.dat
```

The caret (^) character is the EFS quote or escape character. (For more information about escape characters and name and type delimiters, refer to the *OpenVMS User's Manual*.) When this file name is retrieved, it is output to the Compaq C RTL caller just as it was input, with quoted characters unquoted.

Note that, under normal Compaq C RTL parsing rules, an OpenVMS version field in a UNIX style input file name is passed directly to the file system. Therefore, the following example accesses version number 3 of the file with name `data` and type `dat`:

```
data.dat;3
```

With the new feature, this file name is passed to the OpenVMS file system as the following (a file with name `data` and type `dat;3`):

```
data.dat^;3
```

The OpenVMS version number associated with the file is not returned on output.

The scope of this feature applies to allowable characters in both the directory and file-name fields of a UNIX style specification. Enabling this feature does not affect other features of the Compaq C RTL UNIX-to-OpenVMS and OpenVMS-to-UNIX path conversion. If logical-name expansion of the first element in the UNIX specification is enabled, the Compaq C RTL does not quote any special characters that appear in the logical-name equivalence string before presentation to the file system.

5.3.4 POSIX Root Support

Support has been added to provide OpenVMS with a UNIX style root that behaves like a real directory. This allows such actions as the following:

```
% cd /
% mkdir /dirname
% tar -xvf tarfile.tar /dirname
% ls /
```

Programming Features

5.3 Compaq C Run-Time Library Enhancements

Previously, the C RTL did not recognize the slash (/) character as a directory name. The normal processing for a file path starting with the slash character was to interpret the first element as a logical name or device name. If this failed, there was special processing for the name /dev/null and names starting with /bin and /tmp:

```
/dev/null      NLA0:
/bin           SYS$SYSTEM:
/tmp          SYS$SCRATCH:
```

These behaviors are retained for compatibility purposes. In addition, support has been added to the C RTL for the logical name SYS\$POSIX_ROOT as an equivalent to the slash character.

To enable this feature for use by the C RTL, define SYS\$POSIX_ROOT as a concealed logical name. For example:

```
$ DEFINE/TRANSLATION=(CONCEALED,TERMINAL) SYS$POSIX_ROOT "$1$DKA0:[SYS0.abc.]"
```

To disable this feature, enter the following command:

```
$ DEFINE DECC$DISABLE_POSIX_ROOT {1, ENABLE 0, DISABLE}
```

Enabling SYS\$POSIX_ROOT results in the following behavior:

- If the existing translation of a UNIX path starting with the slash (/) character fails and SYS\$POSIX_ROOT is defined, the name is interpreted as if it starts with /sys\$posix_root.
- When converting from an OpenVMS to a UNIX style file name, and the OpenVMS name starts with SYS\$POSIX_ROOT:, then the SYS\$POSIX_ROOT: portion is removed. For example, SYS\$POSIX_ROOT:[dirname] becomes /dirname. If the resulting name could be interpreted as a logical name or as one of the special cases listed, the result is ./dirname instead of /dirname.

5.3.5 32-Bit Group Identifiers

The C RTL supports 32-bit user identification (UID) and group identification (GID). 32-bit UID and GID support was added to some versions of the OpenVMS operating system to provide support for POSIX style identifiers.

Although OpenVMS 7.3-1 does not support POSIX style IDs, it does support 32-bit UID and GID. When an application is compiled to use 32-bit UID/GID, the UID and GID are derived from the UIC, as in previous versions of the operating system. In some cases, such as with the getgroups function, more information might be returned when the application supports 32-bit GIDs.

To compile an application for 32-bit UID or GID support, define the macro `__USE_LONG_GID_T`. To compile an application for 16-bit UID or GID support, define the macro `_DECC_SHORT_GID_T`.

In addition, although you cannot compile with POSIX style IDs enabled on OpenVMS Version 7.3-1, you can run programs that were compiled with POSIX style IDs enabled on OpenVMS systems that do provide that support.

5.3.6 New and Changed C RTL Functions

The Compaq C RTL includes the following new or modified functions for OpenVMS Version 7.3–1:

- New feature switch-setting functions:

```
decc$feature_get_index
decc$feature_get_name
decc$feature_get_value
decc$feature_set_value
```

- New scanf functions:

```
vfscanf      vfwscanf
vscanf      vwscanf
vsscanf     vswscanf
```

- New functions to support 64-bit file sizes and offsets:

```
fseeko
ftello
```

- Functions modified to support 64-bit file sizes and offsets:

```
fstat
ftruncate
ftw
lseek
mmap
stat
truncate
```

5.4 DECdtm System Services

DECdtm provides basic infrastructure for a distributed transaction processing system. Several DECdtm system services are now documented. For information about the XA capability with DECdtm, see Section 4.24.

Table 5–1 lists all newly documented DECdtm system services.

Table 5–1 DECdtm System Services

System Service	System Service with Wait	Description
\$ABORT_TRANS	\$ABORT_TRANSW	Aborts a transaction.
\$ACK_EVENT		Acknowledges an event reported to an RM participant or an RMI.
\$ADD_BRANCH	\$ADD_BRANCHW	Authorizes a new branch to be added to a transaction.
\$CREATE_UID		Generates a universally unique identifier.
\$DECLARE_RM	\$DECLARE_RMW	Creates a new resource manager instance (RMI).
\$END_BRANCH	\$END_BRANCHW	Removes a branch from a transaction and returns the transaction outcome.
\$END_TRANS	\$ENDTRANSW	Ends a transaction.

(continued on next page)

Programming Features

5.4 DECdtm System Services

Table 5–1 (Cont.) DECdtm System Services

System Service	System Service with Wait	Description
\$FORGET_RM	\$FORGETRMW	Deletes an RMI from the calling process.
\$GET_DEFAULT_TRANS		Returns the default transaction of the calling process.
\$GETDTI	\$GETDTIW	Returns information about the state of individual transactions.
\$JOIN_RM	\$JOIN_RMW	Adds a new RM participant to a transaction.
\$SET_DEFAULT_TRANS	\$SET_DEFAULT_TRANSW	Sets or clears the default transaction of the calling process.
\$SETDTI	\$SETDTIW	Sets process current transaction and removes resource managers from transactions.
\$START_BRANCH	\$START_BRANCHW	Adds a new branch to a transaction.
\$START_TRANS	\$START_TRANSW	Starts a transaction.
\$TRANS_EVENT	\$TRANS_EVENTW	Forces certain transaction state changes.

For More Information On	See
DECdtm system services	<i>OpenVMS System Services Reference Manual</i>
Managing DECdtm and XA transactions	<i>OpenVMS System Manager's Manual</i>
XGCP Utility used with DECdtm/XA	<i>OpenVMS System Management Utilities Reference Manual</i>
Programming with DECdtm and XA	<i>OpenVMS Programming Concepts Manual, Volume II</i>

5.5 Open Source Security for OpenVMS—CDSA and SSL

Common Data Security Architecture (CDSA) is a multiplatform, industry-standard security infrastructure. It provides a standards-based, stable programming interface that applications can use to access operating system security services, allowing developers to create cross-platform, security-enabled applications. Applications request security services, such as cryptography and other public key operations, through a dynamically extensible application programming interface (API). These requests are serviced by a set of plug-in security service provider interfaces (SPIs), which can be supplemented or changed as business needs and technologies evolve.

Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha provides industry standard tools that enable secure and accurate network communications with OpenVMS systems in the enterprise. Using strong cryptography and sophisticated hashing algorithms, messages are encrypted for secure transmission and the integrity of the data is validated, ensuring that all data arrived intact and unmodified.

5.5 Open Source Security for OpenVMS—CDSA and SSL

Secure Sockets Layer (SSL) is the open standard security protocol for the secure transfer of sensitive information over the Internet. SSL provides three major capabilities:

- Privacy through encryption
- Server authentication
- Message integrity

Client authentication is available as an optional function.

Compaq SSL V1.0 for OpenVMS Alpha ports the SSL 0.9.6B toolkit and cryptography library to OpenVMS Alpha Version 7.3–1. Compaq SSL for OpenVMS Alpha provides a certificate tool that is a simple interface for viewing and creating SSL certificates. The SSL Certificate Tool enables you to perform the most important certification functions with ease:

- View certificates and certificate requests
- Create certificate requests
- Sign your own certificate
- Create your own certificate authority
- Sign client certificate requests

Additional hash functions are included.

For more detailed information, refer to the *Open Source Security for OpenVMS Alpha, Volume 1: Common Data Security Architecture* or *Open Source Security for OpenVMS Alpha, Volume 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha*.

5.6 Using LDAP with Compaq SSL for OpenVMS

Customers who need added security protection for their enterprise information stored in an LDAP directory can now use Compaq SSL for OpenVMS Alpha in an LDAP session. SSL uses X.509 public key technology to provide the following security functions:

- Integrity and confidentiality of the LDAP dialogue
- Authentication of the client
- Authentication of the server

When you connect to an LDAP directory, you need to log in to use certain services. This involves sending a bind() request to the server. The username/password in a simple bind is sent in clear text. SSL is important because, with Compaq SSL, this bind request is encrypted making the password unreadable except to the server.

For details, refer to the LDAP chapter in the *OpenVMS Utility Routines Manual*.

Programming Features

5.7 New Alpha Floating-Point Registers and Execution Data Block (FRED)

5.7 New Alpha Floating-Point Registers and Execution Data Block (FRED)

A new structure, the floating-point registers and execution data block (FRED), is a combined structure that contains the HWPCB, the floating-point register save area, and the per-kernel thread data. Prior to Version 7.2, OpenVMS supported 16 kernel threads per process. With Version 7.2 and later, OpenVMS supported 256 kernel threads per process. Also, prior to Version 7.3–1, OpenVMS allocated the maximum number of FRED blocks for a given process when that process was created, even if the process did not become multithreaded. With Version 7.3–1, OpenVMS allocates all FRED blocks on an as-needed basis. This results in improved scaling and performance for applications using kernel threads.

For more information, refer to the *OpenVMS Programming Concepts Manual, Volume I*.

5.8 Future Architecture-Specific Code Changes

Note

This information pertains to a future version of OpenVMS and does NOT affect OpenVMS Version 7.3–1.

With the upcoming port of the OpenVMS operating system to the Intel Itanium processor architecture, customers should be aware of conflicts arising from the use of the SYI\$HW_MODEL system service to determine the processor architecture. Previously, HW_MODEL 1024 and greater indicated an Alpha system. With the Itanium-based systems, this will no longer be true.

Compaq recommends that customers determine the system architecture by using either the SYI\$ARCH_TYPE or SYI\$ARCH_NAME system services on all platforms.

The following table lists the specific system services and their current return values or working suggestions:

Processor	Value Returned	
System Marketing Model (SMM), SYI\$HW_MODEL, word field¹		
VAX	1:1023	VAX\$K_name highest issued: 541
Alpha	1024:4095	ALPHA\$K_name highest issued: 2006
Itanium Processor	4096:8191	IPF\$K_name
CPQ Reserved	8191:32767	
ISV Reserved	32768:65535	
SYI\$ARCH_NAME, fifteen characters		
VAX	"VAX"	
Alpha	"Alpha"	

¹This value also returned in high word of SYI\$_NODE_HWVERS

Programming Features

5.8 Future Architecture-Specific Code Changes

Processor	Value Returned
SYIS_ARCH_NAME, fifteen characters	
Itanium Processor	"IA-64"
OTHER	"OTHER"
SYIS_ARCH_TYPE, longword	
OTHER	0
VAX	1
Alpha	2
Itanium Processor	3
SYIS_NODE_SWTYPE, longword	
All	"VMS"
SYIS_CPU, longword	
VAX	CPU code
Others	SS\$_NOTAVAX
SYIS_CPUTYPE (and SYIS_REAL_CPUTYPE), longword	
Alpha	Alpha microprocessor core code
Others	SS\$_NOTALPHA (new)
SYIS_CPULID (new) (and SYIS_REAL_CPULID), longword	
Itanium Processor	Itanium Processor LID value
Others	SS\$_NOTIA64 (new)
SYIS_SID, longword	
VAX	CPU code
Others	256
SYIS_CPULID (new) (and SYIS_REAL_CPULID), longword	
Itanium Processor	Itanium Processor LID value
Others	SS\$_NOTIA64 (new)
SYIS_SID, longword	
VAX	CPU code
Others	256

Programming Features

5.8 Future Architecture-Specific Code Changes

Processor	Value Returned
SYI\$_SYSTYPE, longword	
VAX	Not implemented on VAX
Alpha	Alpha platform family code 1:63 highest issued: 40
Itanium Processor	Itanium Processor platform family code (64:255)
256:32767	Compaq reserved
32768:-0	ISV reserved
SYI\$_XCPU, longword	
All	Implementation-specific
SYI\$_XSID, longword	
All	Implementation-specific

5.9 Support for Image Sections in Galaxy Shared Memory

OpenVMS Galaxy offers the ability to share data between Galaxy instances, (provided by OpenVMS as Galaxy global sections). OpenVMS Version 7.3–1 adds support in the INSTALL utility and the image activator to place write shared image sections in Galaxy global sections. This feature allows existing applications that share data using writeable image sections to be extended to a Galaxy environment.

INSTALL/ADD WRITABLE=GALAXY places write shared image sections in Galaxy global sections. You can also use the IDENT keyword with GALAXY to include the image identity in the name of the Galaxy global section. Doing this lets you use multiple versions of an image simultaneously in a Galaxy system.

Refer to the *OpenVMS Alpha Partitioning and Galaxy Guide* for more details.

5.10 Larger Mailbox Buffer Quotas

Prior to Version 7.3–1, the maximum size for the mailbox buffer quota was 64,000 bytes. OpenVMS Alpha Version 7.3–1 supports the creation of mailboxes with larger buffer quotas for improved application scaling.

When creating a mailbox with the \$CREMBX system service, you can explicitly supply a mailbox buffer quota with the BUFQUO parameter. If not, the mailbox buffer quota is implicitly supplied by the system parameter DEFMBXBUFQUO.

Note the following:

- Temporary mailboxes do not have a specific size limit.
- The size of the mailbox is subtracted from the BYTLM quota of the process.
- Permanent mailboxes do not consume process quota; however, any data in a mailbox consumes nonpaged pool.

5.11 RMS New Features

The following sections describe new RMS features added to the OpenVMS Alpha Version 7.3–1 operating system that improve I/O performance and enable easier porting from UNIX to OpenVMS Alpha.

5.11.1 Increase in Default I/O Transfer Size to Reduce I/O

The multiblock count for a sequential disk file sizes the RMS intermediate buffer for disk block transfers; this count serves as the I/O transfer size. The system default multiblock count is increased from 16 to 32. This increase reduces sequential file read or write I/O requests by a factor of 2. If the application does not assign the multiblock count explicitly (using the RAB\$B_MBC setting) or as a process default (using the DCL command SET RMS_DEFAULT), the new system default multiblock count of 32 applies.

The process or system multiblock count default parameters (set with the SET RMS_DEFAULT/BLOCK_COUNT or SET RMS_DEFAULT/SYSTEM/BLOCK_COUNT command, or with or the SYSGEN parameter RMS_DFMBC) can dynamically restore the default value for a process or system to the former value of 16 blocks. You can also increase the multiblock count to greater than 32 at either the process or system level, after making appropriate changes to working set quotas. The maximum multiblock count is 127 blocks. Because a larger RMS buffer can lead to increased page faulting by a process with a small working set quota, the increase in the system default is limited to 32.

Because files opened as process-permanent files (PPF) by DCL are not affected by process or system multiblock defaults, DCL file opens are not affected by this improvement. However, note that the change to the system multiblock count default can affect any privileged application that uses the FAB\$V_PPF (undocumented) option to open a process-permanent file, if the application does not assign the multiblock count explicitly in the application itself (using the RAB\$B_MBC setting).

While this multiblock increase is compatible in a mixed cluster, it has an indirect effect on VAX or older Alpha versions. If a shared sequential file is first opened from the new release, the system multiblock default is 32 blocks. Because the first accessor determines the multiblock count for all subsequent accessors, 32 is the multiblock count for any subsequent accessors opening the file from an older system.

This new feature also indirectly changes the default extend quantity, which uses 64 blocks (two times the multiblock count of 32) instead of the current 32 blocks (two times the default multiblock count of 16).

5.11.2 Support for POSIX-Compliant File Timestamps

To enable easier porting of UNIX applications to OpenVMS Alpha, RMS adds support for POSIX-compliant file timestamps. The XABDAT structure has been extended to include the POSIX-compliant access date, attribute modification date, and data modification date (\$OPEN and \$DISPLAY). This mechanism can also be used to set all dates on a \$CREATE, which is likely to happen in the case of a file restoration. Support for these dates is restricted to ODS-5 volumes and requires the volume to be enabled for access date support.

Updating the POSIX-compliant access date, attribute modification date, and data modification date values through RMS is supported using the XABITM structure. XABITM\$_NORECORD can be used to suppress update of access time.

Programming Features

5.11 RMS New Features

5.11.3 New System RMS Write-Behind Performance Option

An existing user write-behind (RAB\$V_WBH setting in RAB\$L_ROP) option is available for unshared sequential files to request that RMS overlap writes to disk. Specifying the write-behind option ensures that at least two buffers are allocated by RMS at connect time. When one buffer has been filled with records, RMS switches to the second buffer and issues an asynchronous QIO to write the contents of the first buffer out to disk. RMS does not stall the process, but rather continues to process records in the second buffer while the QIO for the first buffer is completing. Overlapping I/O can be a big performance win for some applications.

OpenVMS Alpha Version 7.3–1 introduces a new system RMS write-behind performance option as a dynamic SYSGEN parameter (RMS_SEQFILE_WBH) to externally request RMS to implement the write-behind feature as a system default (refer to the *OpenVMS System Management Utilities Reference Manual*). If this system option is set, RMS implements the existing RMS user write-behind option as a system default (regardless of the RAB\$L_ROP RAB\$V_WBH setting) whenever write access is requested for an unshared sequential file that is opened for image I/O.

This feature is a system option. It is not the default, however, for the following reasons:

- Enabling write-behind involves a change in error reporting that could affect some applications. (A write error might be reported for a subsequent operation rather than for the write on which the error occurred.)
- The allocation of an additional RMS intermediate buffer might result in more page faulting if process working set quotas are insufficient.

5.12 SDA New Features

OpenVMS Alpha Version 7.3–1 performs a faster large memory dump that significantly reduces the time to write a system crash dump file.

There are also a number of new commands and qualifiers for the System Dump Analyzer (SDA), including the new SDA Extended File Cache (XFC) commands. For more information, refer to the *OpenVMS Alpha System Analysis Tools Manual*.

5.12.1 New VALIDATE TQE Command

The new VALIDATE TQE command validates all the data structures associated with timer queue entries (TQEs).

5.12.2 New TQE Type

The SHOW TQE command displays timer queue entries (TQEs), including the following new TQE type:

Column	Symbol	Description
6	N	TQE should not be deallocated at AST completion.
	–	TQE should be deallocated at AST completion.

5.12.3 New Qualifiers

Table 5–2 describes the new SDA command qualifiers in OpenVMS Version 7.3–1.

Table 5–2 New SDA Command Qualifiers

Command	Qualifier	Description
SHOW PROCESS	/POOL	Displays the dynamic storage pool of the process P0 (process) or P1 (control) region. The following qualifiers from the SHOW PROCESS command now also apply to the SHOW PROCESS/POOL command: /FREE, /HEADER, /MAXIMUM_BYTES, /STATISTICS, /SUBTYPE, /SUMMARY, /TYPE, and /UNUSED.
	/TQE	Displays all timer queue entries associated with the current process.
SHOW RESOURCES	/LIST	Displays summary information for each resource, followed by a list of all locks associated with the resource.

5.12.4 New SDA Extended File Cache (XFC) Commands

The SDA Extended File Cache (XFC) commands along with their parameters and qualifiers are described in Table 5–3. These commands provide analysis tools for tuning the extended file cache to improve the performance of your programs.

Table 5–3 New SDA Extended File Cache (XFC) Commands

Command	Function
EXIT	Exits the XFC SDA extensions. This command has no parameters or qualifiers.
LOAD DSF	Loads the symbols in the specified debug symbol file (DSF) for use with the SDA> FORMAT command. This command has no parameters or qualifiers.
SHOW CONTEXT	Displays the contents of an XFC context structure (CTX). It has one parameter, address , which is the address of the CTX. If no address is supplied, then all the context structures are displayed. It has three qualifiers: <ul style="list-style-type: none"> • /BRIEF Displays a brief summary for each context; for example, the I/O type, start VBN, and length of I/O. • /FULL Displays the complete context structure. This is the default. • /STALLING Displays only contexts that are stalling; for example, those that have a stall reason code other than estrNotStalling.
SHOW EXTENT	Displays the contents of an extent control block (ECB). It has one parameter, address , which is the address of the ECB. It has no qualifiers.

(continued on next page)

Programming Features

5.12 SDA New Features

Table 5–3 (Cont.) New SDA Extended File Cache (XFC) Commands

Command	Function
SHOW FILE	<p>Displays the contents of the cache file block (CFB). It has one parameter, address, which is the address of the CFB. The /OPEN and /CLOSED qualifiers, if present, are ignored. If no address is supplied, then all the CFBs are displayed. It has eight qualifiers:</p> <ul style="list-style-type: none"> • /BRIEF Displays summary information for each cache file block (CFB), such as the CFB address, control volume block (CVB) address, access count, active I/O count, and file ID. • /CLOSED Displays only CFBs whose access count is zero. • /CVB=<i>address</i> Displays only information about any files matching the given Cache Volume Block address. • /EXTENTS Displays the cache extents held in cache for any displayed files. This shows the primary and secondary cache extents along with their data state and VBNs. It also shows a summary of memory usage (pagelets used and pagelets valid) for any displayed files. The /EXTENTS qualifier is incompatible with the /BRIEF qualifier. • /FULL Displays all fields for each Cache File Block. This is the default. • /ID=<i>file-id</i> Displays only information about any files matching the given file-identification. The file ID (FID) is the hexadecimal file-number component in a (file-number, sequence number, relative volume number) format file ID. • /OPEN Displays only CFBs whose access count is greater than zero. • STATISTICS Displays more statistics about the specified file. The /STATISTICS qualifier is incompatible with the /BRIEF qualifier.
SHOW MEMORY	<p>Displays information about memory used by the cache. It has no parameters. It has two qualifiers:</p> <ul style="list-style-type: none"> • /BRIEF Displays summary statistics on XFC memory use. • /FULL Displays full statistics on XFC memory use.
SHOW SUMMARY	<p>Displays general information about the Extended File Cache. It has no parameters. It has one qualifier, /STATISTICS, which displays read and write activity arranged by I/O size.</p>

(continued on next page)

Table 5–3 (Cont.) New SDA Extended File Cache (XFC) Commands

Command	Function
SHOW TABLES	<p>Displays both the extent hash table (EHT) and the file hash table (FHT). It has no parameters. It has four qualifiers:</p> <ul style="list-style-type: none"> • /ALL Displays the contents of the EHT and FHT. This is the default. • /EXTENT Displays the contents of the EHT. • /FILE Displays the contents of the FHT. • /SUMMARY Displays summary information about EHT and FHT.
SHOW TRACE	<p>Displays all or selected portions of the XFC trace buffer, starting with the most recent entry and moving backward in time. It has no parameters. It has six qualifiers:</p> <ul style="list-style-type: none"> • /ALL Displays the entire trace buffer. This is the default. • /CONTAINING=<i>value</i> Displays only records where any of the traced parameters are equal to <i>value</i>. • /CPU=<i>cpu-num</i> Displays only records from threads executing on CPU <i>cpu-num</i>. • /LINENUMBER=<i>linenumber</i> Displays only records from tracepoints at line <i>linenumber</i> in the relevant source files. • /MATCH [=AND OR] Alters the sense of the match condition when more than one of the filter qualifiers /CPU, /LINENUMBER, /FILENAME, /Px, or /CONTAINING are specified. • /Px=<i>value</i> Displays only records where one of the traced parameters P1, P2, P3, or P4 are equal to <i>value</i>.

(continued on next page)

Programming Features

5.12 SDA New Features

Table 5–3 (Cont.) New SDA Extended File Cache (XFC) Commands

Command	Function
SHOW VOLUME	<p>Displays the contents of a cache volume block (CVB). It has one parameter, address, which is the address of a CVB. If no address is supplied, then all volumes are displayed. It has four qualifiers:</p> <ul style="list-style-type: none"> • /BRIEF Displays summary information for each volume. • /FULL Displays a complete list of information about the volume(s). This is the default. • /NAME=DISK\$volume_label Displays information for the volume with the specified name. • /STATISTICS Displays the read and write I/O activity for this volume. The /STATISTICS qualifier is incompatible with the /BRIEF qualifier.

5.13 New Qualifiers for SDA Spinlock Tracing Utility

The SHOW TRACE and SHOW COLLECT commands from the SDA Spinlock Tracing utility have the following new qualifiers:

Qualifier	Description
/RATES	Reports activity as a rate per second and hold/spin time as a percentage of time. This is the default.
/TOTALS	Reports activity as a count and hold/spin time as cycles.

5.14 New and Updated System Services

This section describes new and updated system services for OpenVMS Version 7.3–1. The following table summarizes these changes.

For additional information, refer to the *OpenVMS System Services Reference Manual*.

System Service/Project	Documentation Update
DECdtm	<p>The following services have been added: \$ACK_EVENT, \$ADD_BRANCH[W], \$CREATE_UID, \$DECLARE_RM[W], \$END_BRANCH[W], \$FORGET_RM[W], \$GETDTI[W], \$GET_DEFAULT_TRANS, \$JOIN_RM[W], \$SETDTI[W], \$SET_DEFAULT_TRANS[W], \$START_BRANCH[W], \$TRANS_EVENT[W]</p> <p>The following services have been updated: \$ABORT_TRANS[W], \$END_TRANS[W], \$START_TRANS[W]</p>
\$ACM[W]	New services for Authentication and Credentials Management (ACM).

Programming Features

5.14 New and Updated System Services

System Service/Project	Documentation Update
\$ASSIGN	A new status code, SS\$_EXBYTLM, has been added.
\$CPU_TRANSITION	Two symbolic names, CST\$K_CPU_POWER_OFF and CST\$K_CPU_POWER_ON, have been added to the tran_code argument.
\$CREMBX	The text has been updated for the SS\$_BADPARAM condition value and the bufquo and maxmsg arguments. Note that the maximum value limit of 60,000 bytes for the bufquo argument no longer applies.
\$DELPRC	A new argument, flag , has been added.
\$ERAPAT	The code example has been updated.
\$GETDVI	Two new item codes, DVI\$_MPDEV_CURRENT_PATH and DVI\$_MULTIPATH, have been added. The following bits have been added to the DVI\$_STS item code: UCB\$_V_ALTBSY, UCB\$_V_CLUTRAN, UCB\$_V_DISMOUNT, UCB\$_V_EXFUNC_SUPP, UCB\$_V_FAST_PATH, UCB\$_V_FP_HWINT, UCB\$_V_IOPOST_LOCAL, UCB\$_V_LCL_VALID, UCB\$_V_MNTVERPND, UCB\$_V_NO_ASSIGN, UCB\$_V_PATHVERIP, UCB\$_V_SNAPSHOT, UCB\$_V_SUPMVMMSG, UCB\$_V_SVPN_END, UCB\$_V_WRTLOCKMV.
\$GETTIM	The Description section has been updated.
\$GETJPI	The item code description for JPI\$_ASTEN has been updated to include all processes. The following persona item codes have been added: JPI\$_CLASSIFICATION, JPI\$_PERSONA_ID, JPI\$_PERSONA_AUTHPRIV, JPI\$_PERSONA_PERMPRIV, JPI\$_PERSONA_WORKPRIV, JPI\$_IMAGE_WORKPRIV, JPI\$_IMAGE_AUTHPRIV, JPI\$_IMAGE_PERMPRIV, JPI\$_PERSONA_RIGHTS, JPI\$_SUBSYSTEM_RIGHTS, JPI\$_INSTALL_RIGHTS, JPI\$_PERSONA_RIGHTS_SIZE, JPI\$_SUBSYSTEM_RIGHTS_SIZE, JPI\$_SYSTEM_RIGHTS_SIZE, JPI\$_INSTALL_RIGHTS_SIZE
\$GETQUI	The following items have been added for NUMA RAD Batch Job Support: <ul style="list-style-type: none"> • QUI\$_RAD item code • QUI\$_V_JOB_RAD symbolic name for the QUI\$_JOB_FLAGS item code • QUI\$_V_QUEUE_RAD symbolic name for the QUI\$_QUEUE_FLAGS item code <p>The QUI\$_TRANSLATE_QUEUE function code has been added to the QUI\$_QUEUE_NAME item code list.</p> <p>The string length limit has been changed to 255 characters for the QUI\$_DEVICE_NAME item code.</p>

Programming Features

5.14 New and Updated System Services

System Service/Project	Documentation Update
\$GETSYI	The SYI\$_NODE_HWVERS item code text has been updated. A new error status, SS\$_UNREACHABLE, has been added. The following item codes have been added: SYI\$_ACTIVE_CPU_MASK, SYI\$_AVAIL_CPU_MASK, SYI\$_CPU_AUTOSTART, SYI\$_CPU_FAILOVER, SYI\$_CPUCAP_MASK, SYI\$_GLX_FORMATION, SYI\$_GLX_MAX_MEMBERS, SYI\$_GLX_MBR_MEMBER, SYI\$_GLX_MBR_NAME, SYI\$_GLX_TERMINATION, SYI\$_HP_ACTIVE_CPU_CNT, SYI\$_HP_ACTIVE_SP_CNT, SYI\$_HP_CONFIG_SBB_CNT, SYI\$_HP_CONFIG_SP_CNT, SYI\$_MAX_CPUS, SYI\$_POTENTIAL_CPU_MASK, SYI\$_POTENTIALCPU_CNT, SYI\$_POWERED_CPU_MASK, SYI\$_POWEREDCPU_CNT, SYI\$_PRESENT_CPU_MASK, SYI\$_PRESENTCPU_CNT, SYI\$_PRIMARY_CPUID
\$GETUAI	The context argument restriction for VAX only has been removed.
\$ICC_CONNECT	A new status code, SS\$_NOPRIV, has been added.
\$PROCESS_AFFINITY	A new item code, CAP\$_M_PURGE_WS_IF_NEW_RAD, has been added.
\$PROCESS_CAPABILITIES	A new item code, CAP\$_M_PURGE_WS_IF_NEW_RAD, has been added.
\$PURGWS	A new condition value, SS\$_NOPRIV, has been added.
\$REGISTRY	The item code, REG\$_VALUEDATASIZE, has been added.
\$SNDJBC	The following item codes have been added for NUMA RAD Batch Job Support: SJC\$_RAD and SJC\$_NO_RAD. A new item code, SJC\$_JOB_COMPLETION_STATUS, has been added. The string length limit has been changed to 255 characters for the SJC\$_DEVICE_NAME item code.
\$SUSPND	A cluster-related note has been added to the description section.

5.15 System Services Updates for Batch Job Support for NUMA RADs

The following services have been updated to support batch jobs for NUMA Resource Affinity Domains (RADs):

- \$GETQUI
- \$SNDJBC

For information about the system management updates, see Section 4.6 in this manual, and refer to the *OpenVMS System Services Reference Manual*.

5.15.1 \$GETQUI Updates

This section describes updates to the \$GETQUI service.

5.15 System Services Updates for Batch Job Support for NUMA RADs

5.15.1.1 New \$GETQUI Item Code

The QUI\$_RAD output item code is used to return the value of the RAD attribute attached to a queue or a job. The item code expects a 32-bit buffer in which to write the value of the RAD. A value of -1 is returned to the buffer if no RAD is specified for the queue or the job.

The QUI\$_RAD item code is valid for the following function codes:

- QUI\$_DISPLAY_ENTRY
- QUI\$_DISPLAY_JOB
- QUI\$_DISPLAY_QUEUE

5.15.1.2 New RAD Flag for QUI\$_QUEUE_FLAGS and QUI\$_JOB_FLAGS Item Codes

New bits have been added to the queue flags and job flags item codes for RAD support.

The QUI\$_V_JOB_RAD bit of the QUI\$_JOB_FLAGS item indicates whether the job will use the RAD processing option.

The QUI\$_V_QUEUE_RAD bit of the QUI\$_QUEUE_FLAGS item indicates whether the RAD processing option attribute is specified for the batch queue.

5.15.2 \$SNDJBC Updates

There are two new \$SNDJBC item codes: SJC\$_RAD and SJC\$_NO_RAD.

The SJC\$_RAD input item code is used to specify a RAD number on which to create a batch process. The item accepts a 32-bit integer value as input. The input is validated to be within the range of 0 to SYI\$_RAD_MAX_RADS.

The SJC\$_NO_RAD input item code is a boolean input item code. When the SJC\$_NO_RAD code is specified in a request, the RAD value for the queue or job is removed.

The new item codes are valid for the following function codes:

- SJC\$_ALTER_JOB
- SJC\$_ALTER_QUEUE
- SJC\$_CREATE_JOB
- SJC\$_CREATE_QUEUE
- SJC\$_ENTER_FILE
- SJC\$_START_QUEUE

5.16 NUMA Support Added to Threads Libraries

To improve application scaling and performance on GS Series Alpha (NUMA) systems, a threaded program can now utilize more than one RAD of a NUMA system. For applications with a high degree of parallelism, using multiple RADs can increase performance.

Associated Products Features

This chapter describes significant new features of Compaq OpenVMS operating system associated products. For a listing and directory information about the OpenVMS associated products, refer to the *Guide to OpenVMS Version 7.3-1 CD-ROMs*.

6.1 Compaq Advanced Server for OpenVMS Alpha

Version 7.3 of the Compaq Advanced Server for OpenVMS is supported on OpenVMS Alpha Version 7.3-1. Version 7.3 of the Advanced Server for OpenVMS includes support of the following features:

- Member server role (allowing the server to participate in Windows 2000 native-mode domains)
- Greater compatibility with a wide variety of clients and legacy applications, with support of:
 - Extended character sets, in addition to Extended File Specifications
 - Alias file names, created for shared files whose names do not comply with the more restricted file-naming conventions of legacy applications, such as MS-DOS
- Remote Windows NT printer management (SpoolSS) for printers shared on the Advanced Server for OpenVMS
- DNS for resolving NetBIOS names
- Cluster load balancing using DNS to resolve the server cluster alias name
- PCSI for installing the server
- Windows 2000 client and domain support
- Windows XP Professional client support

To access Advanced Server for OpenVMS, clients must be licensed using the Advanced Server V7.3 license PAK: PWLMXXXCA07.03. For more information, refer to the *Compaq Advanced Server for OpenVMS Guide to Managing Advanced Server Licenses*.

For information about installing Advanced Server for OpenVMS, refer to the *Compaq Advanced Server for OpenVMS Installation and Configuration Guide* provided with the kit documentation.

For information about upgrading from Compaq PATHWORKS for OpenVMS to the Advanced Server for OpenVMS, see *OpenVMS Alpha Version 7.3-1 Release Notes*. For more information on upgrading PATHWORKS for OpenVMS and about installing Advanced Server for OpenVMS, refer to the *Compaq Advanced Server for OpenVMS Installation and Configuration Guide* provided with the kit documentation.

Associated Products Features

6.1 Compaq Advanced Server for OpenVMS Alpha

For information about the latest release of the PATHWORKS for OpenVMS (Advanced Server) product, see Section 6.3.

6.2 Compaq C Run-Time Header Files Library

The Compaq C Run-Time Library provides functions and macros for developing flexible and highly portable applications. To install the header files library from the Compaq C Run-Time Library, do the following:

1. After installing the other Compaq C products and kits, mount the OpenVMS Alpha Version 7.3–1 Layered Products CD-ROM.

2. Read any README files in the following directory:

```
[.CRTL.DOCUMENTATION]
```

3. Copy [.CRTL.KIT]DECC\$RTLDEF.TLB from the CD-ROM to the SYS\$LIBRARY directory. Use a command in the following format:

```
$ COPY device:[CRTL]DECC$RTLDEF.TLB SYS$LIBRARY
```

where *device* is the device name for the CD-ROM drive (for example, DKA0:).

Note

Always copy this file after you have installed the other Compaq C products and kits so that DECC\$RTLDEF.TLB does not get overwritten.

6.3 Compaq PATHWORKS for OpenVMS (Advanced Server)

Version 6.1 of the Compaq PATHWORKS for OpenVMS (Advanced Server) is supported on OpenVMS Version 7.3–1; Version 6.1 of the PATHWORKS product is currently available on the Software Products Library. Earlier versions of PATHWORKS for OpenVMS servers must be upgraded. For more information, see *OpenVMS Alpha Version 7.3–1 Release Notes*.

PATHWORKS for OpenVMS (Advanced Server) is supported on OpenVMS VAX as well as OpenVMS Alpha systems.

New features of Version 6.1 of the PATHWORKS for OpenVMS (Advanced Server) product include the following:

- Member server role (allowing the server to participate in Windows 2000 native-mode domains)
- Greater compatibility with a wide variety of clients and legacy applications, with support of alias file names, created for shared files whose names do not comply with the more restricted file-naming conventions of legacy applications such as MS-DOS
- DNS for resolving NetBIOS names, expanding support of dynamic cluster load balancing to WANs
- Windows 2000 client and domain support
- Windows XP Professional client support

To access PATHWORKS 6.1 for OpenVMS (Advanced Server) on OpenVMS Version 7.3 or 7.3–1, clients must be licensed using the license PAK PWLMXXXCA07.03. For more information, refer to the *Compaq Advanced Server for OpenVMS Guide to Managing Advanced Server Licenses*.

Associated Products Features

6.3 Compaq PATHWORKS for OpenVMS (Advanced Server)

For information about the latest release of Compaq Advanced Server for OpenVMS, see Section 6.1.

6.4 Compaq TCP/IP Services for OpenVMS Version 5.3 New Features

Version 5.3 of TCP/IP Services provides performance improvements and software enhancements, including:

- IPv6 tunneling, providing support for:
 - 6-to-4 tunnelling
 - Anycast address
 - ICMPv6 MIBs
 - An updated API
 - Mobile IPv6 correspondent node
 - Mobile IPv6 in the UNIX management tools
- NTP (Time Service) Version 4
- BIND Version 9 server (on Alpha systems only)
- IMAP mail server
- Kerberos authentication for Telnet client connections
- Subsystem configuration using the SYSCONFIGTAB database (as on UNIX systems)
- Expanded online help for error messages
- Enhancements to the LPD server to improve network printing and printer management in OpenVMS cluster environments
- Support for an editable text file that associates Internet service names and aliases with a port number and protocol (as on UNIX systems)
- Support for OpenVMS Extended File Specifications in the NFS server and the FTP server
- Support for UNIX path names in the FTP server and FTP client (on Alpha systems only)
- Support for the configuration and management of user-written network services (services not supplied with the TCP/IP Services software)

Details on these features and information about upgrading to Version 5.3 of TCP/IP Services for OpenVMS are available in the *Compaq TCP/IP Services for OpenVMS Release Notes*. Additional product information is available at:

<http://www.openvms.compaq.com/network/tcpip.html>

Associated Products Features

6.5 OpenVMS Management Station Version 3.2

6.5 OpenVMS Management Station Version 3.2

OpenVMS Management Station Version 3.2 is included with this release of OpenVMS. OpenVMS Management Station is a powerful Windows based management tool for managing printers, user accounts, and storage devices.

With this version of OpenVMS Management Station, you can manage a wide range of storage devices across multiple OpenVMS Cluster systems. It also provides Fibre Channel disk device support and other enhancements.

For more information, refer to the *OpenVMS Management Station Overview and Release Notes*.

6.6 Powerstorm 300/350 Graphics Support Version 2.0 for OpenVMS

Version 2.0 of this product was released in May 2002. It is supported on OpenVMS Alpha Version 7.2–1 and later. Version 2.0 includes new features for the OpenGL Utility Toolkit (GLUT) and fixes to former problems.

For complete details about Powerstorm 300/350 Graphics Support Version 2.0 for OpenVMS, refer to the release notes that ship with the product. The release notes are supplied in HTML, PDF, and text format and are located in the SYS\$HELP directory after the software kit is installed:

P300_350_V2_README.HTML

P300_350_V2_README.PDF

P300_350_V2_README.TXT

6.7 UNIX Portability Features

Starting with OpenVMS Alpha Version 7.3–1, we are introducing UNIX portability features into the base operating system. We will continue to add more UNIX features into the base operating system over time. To start the UNIX portability effort, we are extending our C Run-Time Library (CRTL) and ODS-5 file system to support UNIX features such as case-sensitive file names, time of last file access, hard links, and more. In addition, we are introducing a UNIX shell and utilities environment with the GNV Early Adopter's Kit on the Open Source Tools for OpenVMS CD-ROM.

6.8 Open Source Tools for OpenVMS CD-ROM

An Open Source Tools¹ for OpenVMS CD-ROM is included with the OpenVMS Version 7.3–1 kit. It is a collection of various ports of Open Source projects to OpenVMS that our Engineering team has facilitated. Please note that these ports are provided as free software under the terms of the GNU Public License.²

The Open Source Tools for OpenVMS include the following:

¹ This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. For more details, see the GNU Public License located in the GNV kit directory.

² You can redistribute it and/or modify it under the terms of the GNU Public License as published by the Free Software Foundation; Version 2.1 of the License.

Associated Products Features

6.8 Open Source Tools for OpenVMS CD-ROM

- **GNV Early Adopter's Kit**

The GNV product is an open source, GNU-based, UNIX-style environment for OpenVMS for UNIX application developers, system managers, and users. It facilitates development and porting of UNIX software to OpenVMS.

GNU is a UNIX-like operating system provided as free software. GNV provides a UNIX-like shell (command-line interpreter) environment and a C Run-Time Library (C RTL) supplemental library that provides utilities typically found on UNIX systems. The shell used by GNV is bash (Bourne-Again Shell, from GNU, using the POSIX.2 specification).
- **CDRECORD sources**

Open Source CDRECORD capability ships as part of OpenVMS Alpha Version 7.3–1. The full source kit used to create the CDRECORD images is provided.
- **Stunnel**

Stunnel is a program that allows you to encrypt arbitrary TCP connections inside an SSL (Secure Sockets Layer) connection from your OpenVMS system to another system. Stunnel also lets you secure non-SSL aware applications (such as Telnet, IMAP, and LDAP) by having Stunnel provide the encryption. Changes to the original application are then not required. Both images and sources are provided.
- **VMSTAR**

VMSTAR is a TAR reader/writer for the OpenVMS operating system. It can read archives ("tarfiles") on disk or tape created by the Un*x "tar" command. It can also create archives.
- **Zip**

Zip is a utility that compresses large or multiple files into a single ZIP archive. UnZip decompresses the files stored in a ZIP archive, restoring them to their original state.
- **OpenVMS Migration Software³**

This utility enables software written for OpenVMS VAX to be executed on an OpenVMS Alpha system. It includes translated sharable images of the runtime libraries from OpenVMS 7.3, native images of the DECmigrate utilities, and much more. Additional information is available at <http://www.softresint.com/AlphaMigrate.htm>
- **SSL sources**

OpenSSL (Secure Socket Layer) ships as an optional Layered Product with OpenVMS Alpha Version 7.3–1. The sources used to build the port of OpenSSL 0.9.6b for OpenVMS are provided.

³ This software is provided as is and is not covered under the GNU Public License. OpenVMS engineering is providing these images as is without any warranty. Sources for this tool are not provided.

Part II

OpenVMS Documentation

OpenVMS Documentation Overview

Table 7–1 describes any reorganization of the OpenVMS documentation set for OpenVMS Version 7.3–1. For this release there are four new manuals and two archived manuals.

Table 7–1 Documentation Set Changes for OpenVMS Version 7.3–1

New Manuals	
<i>COM, Registry, and Events for OpenVMS Developer's Guide</i>	This manual is intended for programmers developing applications that move easily between the OpenVMS and Windows NT environments. Read this manual if you are encapsulating existing OpenVMS applications or data, or creating new COM applications for OpenVMS systems. It also provides information for those who want to use the OpenVMS Registry to store information about their OpenVMS systems alone, or who want to use the OpenVMS Registry as a shared repository for both OpenVMS and Windows NT registry information. This manual was formerly named <i>OpenVMS Connectivity Developer Guide</i> , and is included in the COM for OpenVMS kit and on the COM for OpenVMS Web site at http://www.openvms.compaq.com/openvms/products/dcom/index.html
<i>Compaq C Run-Time Library Reference Manual for OpenVMS Systems</i>	This manual, now included with OpenVMS documentation, is intended for programmers who need reference information on the functions and macros found in the Compaq C RTL that perform I/O operations, character and string manipulation, mathematical operations, error detection, subprocess creation, system access, and screen management. It also notes portability concerns between operating systems, and describes the Compaq C for OpenVMS socket routines used for writing Internet application programs for the TCP/IP protocol.
<i>Open Source Security for OpenVMS Alpha, Volume 1: Common Data Security Architecture</i>	This new manual for Version 7.3–1 is for application developers who want to use the Common Data Security Architecture (CDSA) to add security to their programs. It briefly describes CDSA, gives information about installation and initialization, and provides example programs. It contains the CDSA application programming interface modules.
<i>Open Source Security for OpenVMS Alpha, Volume 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha</i>	This new manual for Version 7.3–1 is for application developers who want to protect communication links to OpenVMS applications with Compaq Secure Sockets Layer (Compaq SSL) for OpenVMS Alpha. It contains installation instructions, release notes, and provides example programs. It includes programming information and a reference section for the OpenSSL application programming interface modules.

(continued on next page)

OpenVMS Documentation Overview

Table 7–1 (Cont.) Documentation Set Changes for OpenVMS Version 7.3–1

Archived Manuals	
<i>OpenVMS Guide to Extended File Specifications</i>	<p>Beginning with Version 7.3–1, this manual is archived. The online version is included with other archived manuals in a separate directory on the OpenVMS documentation CD-ROMs.</p> <p>The information contained in this manual has been distributed to the following manuals in the OpenVMS documentation set:</p> <ul style="list-style-type: none">• <i>OpenVMS User’s Manual</i>• <i>OpenVMS System Manager’s Manual</i>• <i>Guide to OpenVMS File Applications</i>• <i>OpenVMS System Services Reference Manual</i>• <i>OpenVMS Programming Concepts Manual</i>
<i>OpenVMS Master Index</i>	<p>Beginning with Version 7.3–1, this manual is archived. The online version is included with other archived manuals in a separate directory on the OpenVMS documentation CD-ROMs.</p>

OpenVMS Printed and Online Documentation

OpenVMS documentation is provided, as follows:

- Printed documentation
If you need paper documents, you can purchase most OpenVMS manuals as printed books.
- Online documentation on CD-ROM
All OpenVMS manuals are available in online formats on CD-ROMs that also include the documentation for many associated products. You automatically receive the documentation CD-ROMs in your OpenVMS media kit.
- Online documentation on the OpenVMS Web site
You can preview or read any OpenVMS document, including archived manuals, on the OpenVMS Web site.
- Online help
You can quickly display online help for OpenVMS commands, utilities, and system routines when you need task-related information.

The following sections describe each format in which OpenVMS documentation is provided and specifies the titles that are available in that format.

8.1 Printed Documentation

Some printed documentation comes with your OpenVMS media kit. All other printed manuals are separately orderable, either in kits or individually. This section describes the OpenVMS printed documentation offerings, which are broken down into the following categories:

- Media kit
- Documentation sets: Base Set and Full Set
- System integrated products
- Archived manuals

8.1.1 OpenVMS Media Kit Documentation

The OpenVMS media kit contains the documents you need to get started with the latest version of the OpenVMS operating system. Table 8–1 lists the books included in the OpenVMS media kit. The books you receive are determined by whether you are a new or a service customer. New customers receive all books; service customers receive only new books and books that have been updated since the last release. You can also order the manuals listed in Table 8–1 individually.

OpenVMS Printed and Online Documentation

8.1 Printed Documentation

Table 8–1 OpenVMS Media Kit Manuals

Manual	Order Number
<i>OpenVMS License Management Utility Manual</i> ¹	AA-PVXUF-TK
<i>Guide to OpenVMS Alpha Version 7.3–1 CD-ROMs</i>	AA-RSCZA-TE
<i>OpenVMS Alpha Version 7.3–1 New Features and Documentation Overview</i>	AA-RSHYA-TE
<i>OpenVMS Alpha Version 7.3–1 Upgrade and Installation Manual</i>	AA-QSE8E-TE
<i>OpenVMS Alpha Version 7.3–1 Release Notes</i>	AA-RSD0A-TE

¹Not revised for Version 7.3–1

8.1.2 OpenVMS Documentation Sets

OpenVMS documentation is available in the following documentation sets:

Documentation Set	Description	Order Number
Full set	Intended for users who need extensive explanatory information for all major OpenVMS resources. Contains all the OpenVMS documentation in one offering. Includes the Base Documentation set.	QA-001AA-GZ.7.3-1
Base set	Subset of the Full Documentation set. Intended for general users and system managers of small standalone systems. Includes the most commonly used OpenVMS manuals.	QA-09SAA-GZ.7.3-1

In addition to ordering the Full or Base Documentation set, you can order any OpenVMS manual individually.

Table 8–2 lists the manuals in the OpenVMS Base and Full Documentation sets. For a description of each manual, see Section 9.2.

Table 8–2 OpenVMS Full Documentation Set (QA-001AA-GZ.7.3-1)

Manual	Order Number
OpenVMS Base Documentation Set	QA-09SAA-GZ.7.3-1
<i>OpenVMS DCL Dictionary: A–M</i> ¹	AA-PV5KH-TK
<i>OpenVMS DCL Dictionary: N–Z</i> ¹	AA-PV5LH-TK
<i>OpenVMS Guide to System Security</i> ¹	AA-Q2HLF-TE
<i>OpenVMS System Management Utilities Reference Manual: A–L</i> ¹	AA-PV5PG- TK
<i>OpenVMS System Management Utilities Reference Manual: M–Z</i> ¹	AA-PV5QG- TK
<i>OpenVMS System Manager’s Manual, Volume 1: Essentials</i> ¹	AA-PV5MG-TK
<i>OpenVMS System Manager’s Manual, Volume 2: Tuning, Monitoring, and Complex Systems</i> ¹	AA-PV5NG-TK
<i>OpenVMS User’s Manual</i> ¹	AA-PV5JF-TK

¹Revised for Version 7.3–1.

(continued on next page)

Table 8–2 (Cont.) OpenVMS Full Documentation Set (QA-001AA-GZ.7.3-1)

Manual	Order Number
OpenVMS Base Documentation Set	QA-09SAA-GZ.7.3-1
<i>OpenVMS Alpha Version 7.3–1 New Features and Documentation Overview</i> ²	AA-RSHYA-TE
<i>OpenVMS Alpha Version 7.3–1 Release Notes</i> ²	AA-RSD0A-TE
Additional Books Included in the Full Documentation Set	QA-001AA-GZ.7.3-1
<i>Availability Manager User’s Guide</i> ¹	AA-RNSJB-TE
<i>COM, Registry, and Events for OpenVMS Developer’s Guide</i> ²	AA-RSCWA-TE
<i>Compaq C Run-Time Library Reference Manual for OpenVMS Systems</i> ²	AA-PUNEK-TK
<i>Compaq C Run-Time Library Utilities Reference Manual</i>	AA-R238C-TE
<i>Compaq Portable Mathematics Library</i>	AA-PV6VE-TE
<i>DECams User’s Guide</i>	AA-Q3JSE-TE
<i>DEC Text Processing Utility Reference Manual</i>	AA-PWCCD-TE
<i>Extensible Versatile Editor Reference Manual</i>	AA-PWCDD-TE
<i>Guidelines for OpenVMS Cluster Configurations</i> ¹	AA-Q28LF-TK
<i>Guide to Creating OpenVMS Modular Procedures</i>	AA-PV6AD-TK
<i>Guide to OpenVMS File Applications</i> ¹	AA-PV6PE-TK
<i>Guide to the POSIX Threads Library</i>	AA-QSBPD-TE
<i>Guide to the DEC Text Processing Utility</i>	AA-PWCBD-TE
<i>Open Source Security for OpenVMS Alpha, Volume 1: Common Data Security Architecture</i> ²	AA- RSCUA-TE
<i>Open Source Security for OpenVMS Alpha, Volume 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha</i> ²	AA- RSCVA-TE
<i>OpenVMS Alpha Partitioning and Galaxy Guide</i> ¹	AA-REZQD-TE
<i>OpenVMS Alpha Guide to Upgrading Privileged-Code Applications</i>	AA-QSBGD-TE
<i>OpenVMS Alpha System Analysis Tools Manual</i> ¹	A A-REZTC-TE
<i>OpenVMS Calling Standard</i>	AA-QSBBD-TE
<i>OpenVMS Cluster Systems</i> ¹	AA-PV5WF-TK
<i>OpenVMS Command Definition, Librarian, and Message Utilities Manual</i>	AA-QSBDD-TE
<i>OpenVMS Debugger Manual</i>	AA-QSBJD-TE
<i>OpenVMS Delta /XDelta Debugger Manual</i>	AA-PWCAD-TE
<i>OpenVMS I/O User’s Reference Manual</i> ¹	AA-PV6SE-TK
<i>OpenVMS Linker Utility Manual</i>	AA-PV6CD-TK
<i>OpenVMS MACRO-32 Porting and User’s Guide</i>	AA-PV64D-TE
<i>OpenVMS Management Station Overview and Release Notes</i>	AA-QJGCE-TE
<i>OpenVMS Performance Management</i>	AA-R237C-TE
<i>OpenVMS Programming Concepts Manual, Volume I</i> ¹	AA - RNSHB-TK
<i>OpenVMS Programming Concepts Manual, Volume II</i> ¹	A A- PV67F-TK

¹Revised for Version 7.3–1.

²New for Version 7.3–1.

(continued on next page)

OpenVMS Printed and Online Documentation

8.1 Printed Documentation

Table 8–2 (Cont.) OpenVMS Full Documentation Set (QA-001AA-GZ.7.3-1)

Manual	Order Number
Additional Books Included in the Full Documentation Set	QA-001AA-GZ.7.3-1
<i>OpenVMS Record Management Services Reference Manual</i> ¹	AA-PV6RE-TK
<i>OpenVMS Record Management Utilities Reference Manual</i>	AA-PV6QD-TK
<i>OpenVMS RTL General Purpose (OTS\$) Manual</i>	AA-PV6HD-TK
<i>OpenVMS RTL Library (LIB\$) Manual</i>	AA-QSBHD-TE
<i>OpenVMS RTL Screen Management (SMG\$) Manual</i>	AA-PV6LD-TK
<i>OpenVMS RTL String Manipulation (STR\$) Manual</i>	AA-PV6MD-TK
<i>OpenVMS System Messages: Companion Guide for Help Message Users</i>	AA-PV5TD-TK
<i>OpenVMS System Services Reference Manual: A-GETUAI</i> ¹	AA-QSBME-TE
<i>OpenVMS System Services Reference Manual: GETUTC-Z</i> ¹	AA-QSBNE-TE
<i>OpenVMS Utility Routines Manual</i> ¹	AA-PV6EE-TK
<i>OpenVMS VAX RTL Mathematics (MTH\$) Manual</i>	AA-PVXJD-TE
<i>OpenVMS VAX System Dump Analyzer Utility Manual</i>	AA-PV6TD-TE
<i>POLYCENTER Software Installation Utility Developer's Guide</i>	AA-Q28MD-TK
<i>VAX MACRO and Instruction Set Reference Manual</i>	AA-PS6GD-TE
<i>Volume Shadowing for OpenVMS</i> ¹	AA-PVXMH-TE

¹Revised for Version 7.3–1.

8.1.3 Documentation for System Integrated Products

System Integrated Products (SIPs) are included in the OpenVMS software, but you must purchase separate licenses to enable them. Table 8–3 shows the documentation associated with System Integrated Products.

Table 8–3 System Integrated Products Documentation

System Integrated Product	Related Documentation
Compaq Galaxy Software Architecture on OpenVMS Alpha	The documentation is included in the OpenVMS Full Documentation set, or you can order it separately.
OpenVMS Clusters	The OpenVMS Cluster documentation is included in the OpenVMS Full Documentation set.
RMS Journaling for OpenVMS	You must purchase the RMS Journaling documentation separately. Use the following part number: <i>RMS Journaling for OpenVMS Manual: AA-JG41C-TE</i>
Volume Shadowing for OpenVMS	The documentation is included in the OpenVMS Full Documentation set, or you can order it separately.

8.1.4 Archived OpenVMS Documentation

OpenVMS continuously updates, revises, and enhances the OpenVMS operating system documentation. From time to time, manuals are archived. You can access the archived manuals online from the documentation CD-ROMs or from the following Web site:

<http://www.compaq.com/openvms>

For a list of the archived OpenVMS manuals, see Section 9.6.

8.2 Documentation Pilot Project

With Version 7.3–1, OpenVMS is introducing several books that have been authored and published using the Standard Generalized Markup Language (SGML). SGML is an industry standard and will provide many benefits to both the customer and OpenVMS documentation. The following books have been produced with this new tool:

- *Open Source Security for OpenVMS Alpha, Volume 1: Common Data Security Architecture*
- *Open Source Security for OpenVMS Alpha, Volume 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha*
- *OpenVMS Alpha Version 7.3–1 Upgrade and Installation Manual*
- *OpenVMS Guide to System Security*
- *TCP/IP Services for OpenVMS Concepts and Planning*
- *OpenVMS Alpha Partitioning and Galaxy Guide*
- *Guide to OpenVMS Alpha Version 7.3–1 CD-ROMs*

Due to several enhancements to the HTML versions of these manuals, you need the Compaq Secure Web Browser to view the HTML files.

8.3 Online Documentation on CD-ROM

Online documentation for the OpenVMS operating system and many associated products is provided on two CD-ROMs. One is an ISO9660 Level 2 CD-ROM that is readable on Windows and Macintosh systems. The other is a Files-11 CD-ROM that is readable on Windows and OpenVMS systems. The contents of the CD-ROMs are the same, except:

- The ISO9660 Level 2 CD-ROM contains Adobe Acrobat Reader Version 5.0.5.
- The Files-11 CD-ROM contains the Compaq Secure Web Browser for OpenVMS Alpha (based on Mozilla) and a command procedure to launch the browser.

8.3.1 Online Formats

The documentation CD-ROMs contain documentation in a number of formats.

Documentation	Available Formats
Current OpenVMS manuals	HTML, PDF
Archived manuals	PDF
<i>OpenVMS Alpha Version 7.3–1 Upgrade and Installation Manual</i>	HTML ² , PDF, PostScript, ASCII text
<i>OpenVMS Alpha Version 7.3–1 Release Notes</i>	HTML, PDF, PostScript, ASCII text
<i>OpenVMS Alpha Version 7.3–1 New Features and Documentation Overview</i>	HTML, PDF, PostScript, ASCII text
Layered product documents (Release Notes, Install Guides, Software Product Descriptions (SPDs), Cover Letters)	PostScript, ASCII text

²Requires Compaq Secure Web Browser

OpenVMS Printed and Online Documentation

8.3 Online Documentation on CD-ROM

Bookreader files are no longer available on the documentation CD-ROMs.

8.3.2 PDF Reader

The Adobe Acrobat Reader is provided for viewing PDF files. This self-extracting file can be installed on a computer running Windows. It is located on the ISO9660 Level 2 CD-ROM.

For information about how to access documents on the documentation CD-ROMs and about the PDF reader, refer to the *Guide to OpenVMS Alpha Version 7.3-1 CD-ROMs*.

8.4 Online Documentation on the OpenVMS Web Site

You can access OpenVMS manuals in various online formats from the following OpenVMS Web site:

<http://www.compaq.com/openvms>

This site contains links to current versions of manuals in the OpenVMS Full Documentation set as well as to manuals for selected layered products.

8.5 Online Help

The OpenVMS operating system provides online help for the commands, utilities, and system routines documented in the Full Documentation set.

You can use the Help Message facility to quickly access online descriptions of system messages. In addition, you can add your own source files, such as messages documentation that you have written to the Help Message database. The *OpenVMS System Messages: Companion Guide for Help Message Users* manual explains how to use the Help Message facility. You can also access DCL Help for Help Message by entering:

```
$ HELP HELP/MESSAGE
```

Reference information for the following OpenVMS utility routines is now included in online help under RTL_Routines:

- Access Control List (ACL) Editor (ACLEEDIT\$)
- Backup (BACKUP) (BACKUP\$)
- Command Language (CLI) (CLI\$)
- Command File Qualifier (UTIL\$)
- Convert (CONVERT) (CONV\$)
- Data Compression/Expansion (DCX) (DCX\$)
- DEC Text Processing Utility (DECTPU) (TPU\$)
- EDT (EDT\$EDIT; FILEIO; WORKIO; XLATE)
- File Definition Language (FDL) (FDL\$)
- Librarian (LBR) (LBR\$)
- LOGINOUT (LGI) (LGI\$)
- Mail Utility (MAIL) (MAIL\$)
- National Character Set (NCS) (NCS\$)
- Print Symbiont Modification (PSM) (PSM\$; USER-x)
- Symbiont/Job Controller Interface (SMB) (SMB\$)
- Sort/Merge (SOR) (SOR\$)

Descriptions of OpenVMS Manuals

This chapter provides summary descriptions for the following OpenVMS documentation:

- Manuals in the OpenVMS Media Kit (Section 9.1)
- Manuals in the OpenVMS Base and Full Documentation sets (Section 9.2 and Section 9.3)
- OpenVMS Alpha Device Driver manual (Section 9.4)
- RMS Journaling manual (Section 9.5)
- Archived manuals (Section 9.6)

9.1 Manuals in the OpenVMS Media Kit

Guide to OpenVMS Alpha Version 7.3–1 CD-ROMs

Provides information about the OpenVMS Alpha operating system and documentation CD-ROMs. Lists the contents of the OpenVMS Version 7.3-1 CD-ROM package, includes pointers to installation information, and gives instructions about how to access manuals on the documentation CD-ROM.

OpenVMS License Management Utility Manual

Describes the License Management Facility (LMF), the OpenVMS license management tool. LMF includes the License Management Utility (LICENSE) and VMSLICENSE.COM, the command procedure you use to register, manage, and track software licenses.

OpenVMS Alpha Version 7.3–1 Upgrade and Installation Manual

Provides step-by-step instructions for installing and upgrading the OpenVMS Alpha operating system on Alpha computers. Includes information about booting, shutdown, backup, and licensing procedures.

OpenVMS Alpha Version 7.3–1 New Features and Documentation Overview

Describes new and improved components for the Alpha and VAX operating systems for the Version 7.3-1 release. Includes information about OpenVMS documentation changes for Version 7.3-1 as well as the printed and online OpenVMS documentation offerings.

OpenVMS Alpha Version 7.3–1 Release Notes

Describes changes to the software; installation, upgrade, and compatibility information; new and existing software problems and restrictions; and software and documentation corrections.

Descriptions of OpenVMS Manuals

9.2 Manuals in the OpenVMS Base Documentation Set

9.2 Manuals in the OpenVMS Base Documentation Set

OpenVMS DCL Dictionary

Describes the DIGITAL Command Language (DCL) and provides an alphabetical listing of detailed reference information and examples for all DCL commands and lexical functions. This manual is in two volumes.

OpenVMS Guide to System Security

Describes the security features available in the OpenVMS Alpha and VAX operating systems. Explains the purpose and proper application of each feature in the context of specific security needs.

OpenVMS System Management Utilities Reference Manual

Presents reference information about the utilities you can use to perform system management tasks on your system as well as the tools to control and monitor system access and resources. Includes a description of the AUTOGEN command procedure. This manual is in two volumes.

OpenVMS System Manager's Manual, Volume 1: Essentials

Provides instructions for setting up and maintaining routine operations such as starting up the system, installing software, and setting up print and batch queues. Also explains routine disk and magnetic tape operations.

OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems

Describes how to configure and control the network, how to monitor the system, and how to manage system parameters. Also includes information about OpenVMS Cluster systems, network environments, and DECdtm functionality.

OpenVMS User's Manual

Provides an overview of the operating system and presents basic concepts, task information, and reference information that allow you to perform daily computing tasks. Describes how to work with files and directories. Also includes these additional topics:

- Sending messages with the Mail utility and the Phone utility
- Using the Sort/Merge utility
- Using logical names and symbols
- Writing command procedures
- Editing files with the EVE and EDT text editors

OpenVMS Alpha Version 7.3-1 New Features and Documentation Overview

Describes new and improved components for the Alpha and VAX operating systems for the Version 7.3-1 release. Includes information about OpenVMS documentation changes for Version 7.3-1 as well as the printed and online OpenVMS documentation offerings.

OpenVMS Alpha Version 7.3-1 Release Notes

Describes changes to the software; installation, upgrade, and compatibility information; new and existing software problems and restrictions; and software and documentation corrections.

9.3 Additional Manuals in the OpenVMS Full Documentation Set

Availability Manager User's Guide

Describes how to use the Compaq Availability Manager system management tool, from either an OpenVMS Alpha or a Windows node, to monitor one or more OpenVMS nodes on an extended local area network (LAN) or to target a specific node or process for detailed analysis.

COM, Registry, and Events for OpenVMS Developer's Guide

For programmers developing applications that move easily between the OpenVMS and Windows NT environments. Read this manual if you are encapsulating existing OpenVMS applications or data, or creating new COM applications for OpenVMS systems. It also provides information for those who want to use the OpenVMS Registry to store information about their OpenVMS systems alone, or who want to use the OpenVMS Registry as a shared repository for both OpenVMS and Windows NT registry information. This manual was formerly available online as the *OpenVMS Connectivity Developer Guide*.

Compaq C Run-Time Library Reference Manual for OpenVMS Systems

Provides reference information on the functions and macros found in the Compaq C RTL that perform I/O operations, character and string manipulation, mathematical operations, error detection, subprocess creation, system access, and screen management. Includes portability concerns between operating systems, and describes the Compaq C for OpenVMS socket routines used for writing Internet application programs for the TCP/IP protocol.

Compaq C Run-Time Library Utilities Reference Manual

Provides detailed usage and reference information about the Compaq C Run-Time Library utilities for managing localization and time zone data in international software applications.

Compaq Portable Mathematics Library

Documents the mathematics routines in the Compaq Portable Mathematics Library (DPML), supplied only with OpenVMS Alpha systems. VAX programmers should refer to the *OpenVMS VAX RTL Mathematics (MTH\$) Manual*.

DECamds User's Guide

Provides information for installing and using the DECamds software. DECamds is a system management tool that lets you monitor, diagnose, and track events in OpenVMS system and OpenVMS Cluster environments.

DEC Text Processing Utility Reference Manual

Describes the DEC Text Processing Utility (DECTPU) and provides reference information about the EDT Keypad Emulator interfaces to DECTPU.

Extensible Versatile Editor Reference Manual

Contains command reference information about the EVE text editor. Also provides a cross-reference between EDT and EVE commands.

Guidelines for OpenVMS Cluster Configurations

This manual provides information to help you choose systems, interconnects, storage devices, and software. It can help you configure these components to achieve high availability, scalability, performance, and ease of system management. Detailed directions using SCSI and Fibre Channel in an OpenVMS Cluster system are also included in this manual.

Descriptions of OpenVMS Manuals

9.3 Additional Manuals in the OpenVMS Full Documentation Set

Guide to Creating OpenVMS Modular Procedures

Describes how to perform a complex programming task by dividing it into modules and coding each module as a separate procedure.

Guide to OpenVMS File Applications

Contains guidelines for designing, creating, and maintaining efficient data files by using Record Management Services (RMS). This manual is intended for application programmers and designers responsible for programs that use RMS files, especially if performance is an important consideration.

Guide to the POSIX Threads Library

Describes the POSIX Threads Library (formerly named DECthreads) package, Compaq's multithreading run-time libraries. Use the routines in this package to create and control multiple threads of execution within the address space provided by a single process. Offering both usage tips and reference synopses, this document describes three interfaces: routines that conform to the IEEE POSIX 1003.1c standard (called *pthread*), routines that provide thread-related services in nonthreaded applications (called thread-independent services or *tis*), and a set of Compaq proprietary routines (called *cma*) that provide a stable, upwardly compatible interface.

Guide to the DEC Text Processing Utility

Provides an introduction to developing DECTPU programs.

Open Source Security for OpenVMS Alpha, Volume 1: Common Data Security Architecture

For application developers who want to use the Common Data Security Architecture (CDSA) to add security to their programs. Describes CDSA, gives information about installation and initialization, and provides example programs. Contains the CDSA application programming interface modules.

Open Source Security for OpenVMS Alpha, Volume 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha

For application developers who want to protect communication links to OpenVMS applications with Compaq Secure Sockets Layer (Compaq SSL) for OpenVMS Alpha. Contains installation instructions, release notes, and provides example programs. Includes programming information and a reference section for the OpenSSL application programming interface modules.

OpenVMS Alpha Partitioning and Galaxy Guide

Provides complete details about how to use all of the OpenVMS Galaxy features and capabilities available in OpenVMS Alpha Version 7.3. Includes procedures for creating, managing, and using OpenVMS Galaxy computing environments on AlphaServer 8400, 8200, and 4100 systems.

OpenVMS Alpha Guide to Upgrading Privileged-Code Applications

Explains the OpenVMS Alpha Version 7.0 changes that might impact Alpha privileged-code applications and device drivers as a result of the OpenVMS Alpha 64-bit virtual addressing and kernel threads support provided in OpenVMS Alpha Version 7.0.

Privileged-code applications from versions prior to OpenVMS Alpha Version 7.0 might require the source-code changes described in this guide.

Descriptions of OpenVMS Manuals

9.3 Additional Manuals in the OpenVMS Full Documentation Set

OpenVMS Alpha System Analysis Tools Manual

Describes the following system analysis tools in detail, while also providing a summary of the dump off system disk (DOSD) capability and the DELTA/XDELTA debugger:

- System Dump Analysis (SDA)
- System code debugger (SCD)
- System dump debugger (SDD)
- Watchpoint utility

Intended primarily for the system programmer who must investigate the causes of system failures and debug kernel mode code, such as a device driver.

OpenVMS Calling Standard

Documents the calling standard for the OpenVMS Alpha and VAX operating system.

OpenVMS Cluster Systems

Describes procedures and guidelines for configuring and managing OpenVMS Cluster systems. Also describes how to provide high availability, building-block growth, and unified system management across clustered systems.

OpenVMS Command Definition, Librarian, and Message Utilities Manual

Contains descriptive and reference information about the following utilities:

- Command Definition utility
- Librarian utility
- Message utility

OpenVMS Debugger Manual

Explains the features of the OpenVMS Debugger for programmers.

OpenVMS Delta/XDelta Debugger Manual

Describes the Delta/XDelta utility used to debug programs that run in privileged processor mode or at an elevated interrupt priority level.

OpenVMS I/O User's Reference Manual

Contains the information that system programmers need to program I/O operations using the device drivers that are supplied with the operating system.

OpenVMS Linker Utility Manual

Describes how to use the Linker utility to create images that run on OpenVMS systems. Also explains how to control a link operation with link qualifiers and link options.

OpenVMS MACRO-32 Porting and User's Guide

Describes how to port existing VAX MACRO assembly language code to an OpenVMS Alpha system by using the features of the MACRO-32 compiler. Also documents how to use the compiler's 64-bit addressing support.

OpenVMS Management Station Overview and Release Notes

Provides an overview and release notes for OpenVMS Management Station and describes how to get started using the software. OpenVMS Management Station is a powerful, Microsoft Windows based management tool for system managers and others who perform user account and printer management tasks on OpenVMS systems.

Descriptions of OpenVMS Manuals

9.3 Additional Manuals in the OpenVMS Full Documentation Set

OpenVMS Performance Management

Introduces and explains the techniques used to optimize performance on an OpenVMS system.

OpenVMS Programming Concepts Manual

Describes concepts such as process creation, kernel threads and the kernel threads process structure, interprocess communication, process control, data sharing, condition handling, and ASTs. This two-volume manual uses system services, utility routines, and run-time library (RTL) routines to illustrate mechanisms for utilizing OpenVMS features.

OpenVMS Record Management Services Reference Manual

Provides reference and usage information for all programmers who use RMS data files.

OpenVMS Record Management Utilities Reference Manual

Contains descriptive and reference information about the following RMS utilities:

- Analyze/RMS_File utility
- Convert and Convert/Reclaim utilities
- File Definition Language facility

OpenVMS RTL General Purpose (OTS\$) Manual

Documents the general-purpose routines contained in the OTS\$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS RTL Library (LIB\$) Manual

Documents the general-purpose routines contained in the LIB\$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS RTL Screen Management (SMG\$) Manual

Documents the screen management routines contained in the SMG\$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS RTL String Manipulation (STR\$) Manual

Documents the string manipulation routines contained in the STR\$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS System Messages: Companion Guide for Help Message Users

Describes features of the Help Message facility, a tool that you can use to display message descriptions. Describes the HELP/MESSAGE command and qualifiers and also includes detailed information about customizing the Help Message database. Also provides descriptions of messages that can occur when the system and Help Message are not fully operable.

OpenVMS System Services Reference Manual

Presents the set of routines that the operating system uses to control resources, allow process communication, control I/O, and perform other such operating system functions. This manual is in two volumes.

OpenVMS Utility Routines Manual

Describes the routines that allow a program to use the callable interface of selected OpenVMS utilities.

Descriptions of OpenVMS Manuals

9.3 Additional Manuals in the OpenVMS Full Documentation Set

OpenVMS VAX RTL Mathematics (MTH\$) Manual

Documents the mathematics routines contained in the MTH\$ facility of the OpenVMS Run-Time Library, which is relevant only to programmers using OpenVMS VAX. (Alpha programmers should refer to *Compaq Portable Mathematics Library*.)

OpenVMS VAX System Dump Analyzer Utility Manual

Explains how to use the System Dump Analyzer utility to investigate system failures and examine a running OpenVMS VAX system. VAX programmers should refer to this manual; Alpha programmers should refer to the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

POLYCENTER Software Installation Utility Developer's Guide

Describes the procedure and provides guidelines for developing software products that will be installed using the POLYCENTER Software Installation utility. Intended for developers who are designing installation procedures for software products layered on the OpenVMS operating system.

VAX MACRO and Instruction Set Reference Manual

Documents both the assembler directives of VAX MACRO and the VAX instruction set.

Volume Shadowing for OpenVMS

Describes how to provide high data availability with phase II volume shadowing.

9.4 OpenVMS Alpha Device Driver Manual

Writing OpenVMS Alpha Device Drivers in C

Provides definitive information about writing device drivers in the C programming language for devices connected to Alpha processors. *Writing OpenVMS Alpha Device Drivers in C* is available from Digital Press. For more information, access the following Web site:

<http://www.bh.com/digitalpress>

9.5 RMS Journaling Manual

RMS Journaling for OpenVMS Manual

Describes the three types of RMS Journaling as well as other OpenVMS components that support RMS Journaling. This manual also describes the RMS Recovery utility (which is used to recover data saved using journaling), the transaction processing system services, and system management tasks required when using RMS Journaling.

9.6 Archived Manuals

Table 9–1 lists the OpenVMS manuals that have been archived. Please note the following:

- Most archived manuals can be ordered separately in printed format from Compaq.
- Archived manuals are available in various formats (PostScript or PDF) on the documentation CD-ROMs.

Descriptions of OpenVMS Manuals

9.6 Archived Manuals

- Most information from the archived manuals has been incorporated in other documents or online help.

Table 9–1 Archived OpenVMS Manuals

Manual	Order Number
<i>A Comparison of System Management on OpenVMS AXP and OpenVMS VAX</i>	AA-PV71B-TE
<i>Building Dependable Systems: The OpenVMS Approach</i>	AA-PV5YB-TE
<i>Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver</i>	AA-R0Y8A-TE
<i>Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver</i>	AA-Q28TA-TE
<i>Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver</i>	AA-Q28UA-TE
<i>Guide to OpenVMS AXP Performance Management</i>	AA-Q28WA-TE
<i>Guide to OpenVMS Performance Management</i>	AA-PV5XA-TE
<i>Migrating an Application from OpenVMS VAX to OpenVMS Alpha</i>	AA-KSBKB-TE
<i>Migrating an Environment from OpenVMS VAX to OpenVMS Alpha</i>	AA-QSBLA-TE
<i>Migrating to an OpenVMS AXP System: Planning for Migration</i>	AA-PV62A-TE
<i>Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications</i>	AA-PV63A-TE
<i>OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features</i>	AA-QSBCC-TE
<i>OpenVMS Alpha System Dump Analyzer Utility Manual</i>	AA-PV6UC-TE
<i>OpenVMS AXP Device Support: Developer's Guide</i>	AA-Q28SA-TE
<i>OpenVMS AXP Device Support: Reference</i>	AA-Q28PA-TE
<i>OpenVMS Bad Block Locator Utility Manual</i>	AA-PS69A-TE
<i>OpenVMS Compatibility Between VAX and Alpha</i>	AA-PYQ4C-TE
<i>OpenVMS Developer's Guide to VMSINSTAL</i>	AA-PWBXA-TE
<i>OpenVMS DIGITAL Standard Runoff Reference Manual</i>	AA-PS6HA-TE
<i>OpenVMS EDT Reference Manual</i>	AA-PS6KA-TE
<i>OpenVMS Exchange Utility Manual</i>	AA-PS6AA-TE
<i>OpenVMS Glossary</i>	AA-PV5UA-TK
<i>OpenVMS Guide to Extended File Specifications</i>	AA-REZRB-TE
<i>OpenVMS Master Index</i>	AA-QSBSD-TE
<i>OpenVMS National Character Set Utility Manual</i>	AA-PS6FA-TE
<i>OpenVMS Obsolete Features Manual</i>	AA-PS6JA-TE
<i>OpenVMS Programming Environment Manual</i>	AA-PV66B-TK
<i>OpenVMS Programming Interfaces: Calling a System Routine</i>	AA-PV68B-TK
<i>OpenVMS RTL DECTalk (DTK\$) Manual</i>	AA-PS6CA-TE
<i>OpenVMS RTL Parallel Processing (PPL\$) Manual</i>	AA-PV6JA-TK
<i>OpenVMS Software Overview</i>	AA-PVXHB-TE
<i>OpenVMS SUMSLP Utility Manual</i>	AA-PS6EA-TE

(continued on next page)

Table 9–1 (Cont.) Archived OpenVMS Manuals

Manual	Order Number
<i>OpenVMS System Messages and Recovery Procedures Reference Manual: A–L</i>	AA-PVXKA-TE
<i>OpenVMS System Messages and Recovery Procedures Reference Manual: M–Z</i>	AA-PVXLA-TE
<i>OpenVMS Terminal Fallback Utility Manual</i>	AA-PS6BA-TE
<i>OpenVMS VAX Card Reader, Line Printer, and LPA11–K I/O User’s Reference Manual</i>	AA-PVXGA-TE
<i>OpenVMS VAX Device Support Manual</i>	AA-PWC8A-TE
<i>OpenVMS VAX Device Support Reference Manual</i>	AA-PWC9A-TE
<i>OpenVMS VAX Patch Utility Manual</i>	AA-PS6DA-TE
<i>OpenVMS Wide Area Network I/O User’s Reference Manual</i>	AA-PWC7A-TE
<i>PDP-11 TECO User’s Guide</i>	AA-K420B-TC
<i>POLYCENTER Software Installation Utility User’s Guide</i>	AA-Q28NA-TK
<i>TCP/IP Networking on OpenVMS Systems</i>	AA-QJGDB-TE
<i>Standard TECO Text Editor and Corrector for the VAX, PDP-11, PDP-10, and PDP-8</i>	Available only on CD-ROM

Table 9–2 lists the networking manuals and installation supplements that have been archived.

Table 9–2 Archived Networking Manuals and Installation Supplements

Manual	Order Number
<i>DECnet for OpenVMS Guide to Networking</i>	AA-PV5ZA-TK
<i>DECnet for OpenVMS Network Management Utilities</i>	AA-PV61A-TK
<i>DECnet for OpenVMS Networking Manual</i>	AA-PV60A-TK
<i>OpenVMS VAX Upgrade and Installation Supplement: VAX 8820, 8830, 8840</i>	AA-PS6MA-TE
<i>OpenVMS VAX Upgrade and Installation Supplement: VAX 8200, 8250, 8300, 8350</i>	AA-PS6PA-TE
<i>OpenVMS VAX Upgrade and Installation Supplement: VAX 8530, 8550, 8810 (8700), and 8820–N (8800)</i>	AA-PS6QA-TE
<i>OpenVMS VAX Upgrade and Installation Supplement: VAX 8600, 8650</i>	AA-PS6UA-TE
<i>VMS Upgrade and Installation Supplement: VAX-11/780, 785</i>	AA-LB29B-TE
<i>VMS Upgrade and Installation Supplement: VAX-11/750</i>	AA-LB30B-TE

Descriptions of the archived OpenVMS manuals are as follows:

A Comparison of System Management on OpenVMS AXP and OpenVMS VAX

Discusses system management tools, the impact of Alpha page sizes on system management operations, the system directory structure, interoperability issues, and performance information. Designed for system managers who need to learn quickly how to manage an OpenVMS Alpha system.

Descriptions of OpenVMS Manuals

9.6 Archived Manuals

Building Dependable Systems: The OpenVMS Approach

Offers practical information about analyzing the dependability requirements of your business applications and deciding how to use your computing systems to support your dependability goals. This information is complemented by technical summaries of the dependability features of OpenVMS and related hardware and layered software products.

Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver

Describes the procedures for converting a device driver used on OpenVMS VAX to a device driver that runs on OpenVMS Alpha. This book also contains data structures, routines, and macros for maintaining an Alpha driver written in Macro-32.

Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver

Provides information for upgrading a Step 1 device driver (used in earlier versions of OpenVMS AXP) to a Step 2 device driver. A Step 2 device driver is required for OpenVMS AXP Version 6.1.

Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver

Provides information for migrating a device driver used on OpenVMS VAX to a Step 2 device driver used on OpenVMS AXP Version 6.1.

Guide to OpenVMS AXP Performance Management

Introduces and explains the techniques used to optimize performance on an OpenVMS Alpha system.

Guide to OpenVMS Performance Management

Introduces and explains the techniques used to optimize performance on an OpenVMS VAX system.

Migrating an Application from OpenVMS VAX to OpenVMS Alpha

Describes how to create an OpenVMS Alpha version of an OpenVMS VAX application. Provides an overview of the VAX to Alpha migration process and information to help you plan a migration. It discusses the decisions you must make in planning a migration and the ways to get the information you need to make those decisions. In addition, this manual describes the migration methods available so that you can estimate the amount of work required for each method and select the method best suited to a given application.

Migrating an Environment from OpenVMS VAX to OpenVMS Alpha

Describes how to migrate a computing environment from an OpenVMS VAX system to an OpenVMS Alpha system or a mixed-architecture cluster. Provides an overview of the VAX to Alpha migration process and describes the differences in system and network management on VAX and Alpha computers.

Migrating to an OpenVMS AXP System: Planning for Migration

Describes the general characteristics of RISC architectures, compares the Alpha architecture to the VAX architecture, and presents an overview of the migration process and a summary of migration tools provided by Compaq. The information in this manual is intended to help you define the optimal migration strategy for your application.

Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications

Provides detailed technical information for programmers who must migrate high-level language applications to OpenVMS Alpha. Describes how to set up a development environment to facilitate the migration of applications, helps programmers identify application dependencies on elements of the VAX architecture, and introduces compiler features that help resolve these dependencies. Individual sections of this manual discuss specific application dependencies on VAX architectural features, data porting issues (such as alignment concerns), and the process of migrating VAX shareable images.

OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features

Introduces and describes OpenVMS Alpha operating system support for 64-bit virtual addressing and Very Large Memory (VLM). Intended for system and application programmers, this guide highlights the features and benefits of OpenVMS Alpha 64-bit and VLM capabilities. It also describes how to use these features to enhance application programs to support 64-bit addresses and to efficiently harness very large physical memory.

OpenVMS Alpha System Dump Analyzer Utility Manual

Explains how to use the System Dump Analyzer utility to investigate system failures and examine a running OpenVMS Alpha system. Alpha programmers should refer to this manual; VAX programmers should refer to the *OpenVMS VAX System Dump Analyzer Utility Manual*.

OpenVMS AXP Device Support: Developer's Guide

Describes how to write a driver for OpenVMS Alpha for a device not supplied by Compaq.

OpenVMS AXP Device Support: Reference

Provides the reference material for the *Writing OpenVMS Alpha Device Drivers in C* by describing the data structures, macros, and routines used in device-driver programming.

OpenVMS Bad Block Locator Utility Manual

Describes how to use the Bad Block Locator utility to locate bad blocks on older types of media.

OpenVMS Compatibility Between VAX and Alpha

Compares and contrasts OpenVMS on VAX and Alpha computers, focusing on the features provided to end users, system managers, and programmers.

OpenVMS Developer's Guide to VMSINSTAL

Describes the VMSINSTAL command procedure and provides guidelines for designing installation procedures that conform to standards recommended by Compaq. Intended for developers who are designing installation procedures for software products layered on the OpenVMS operating system.

OpenVMS DIGITAL Standard Runoff Reference Manual

Describes the DSR text-formatting utility.

OpenVMS EDT Reference Manual

Contains complete reference information for the EDT editor.

OpenVMS Exchange Utility Manual

Describes how to use the Exchange utility to transfer files between some foreign format volumes and OpenVMS native volumes.

Descriptions of OpenVMS Manuals

9.6 Archived Manuals

OpenVMS Glossary

Defines terms specific to OpenVMS that are used throughout the documentation.

OpenVMS Guide to Extended File Specifications

Provides an overview of Extended File Specifications and describes the overall differences and impact Extended File Specifications introduce to the OpenVMS environment.

OpenVMS Master Index

Offers an edited compilation of indexes from the manuals in the OpenVMS Full Documentation set.

OpenVMS National Character Set Utility Manual

Describes how to use the National character set utility to build NCS definition files.

OpenVMS Obsolete Features Manual

Presents the DCL commands, system services, RTL routines, and utilities made obsolete by VMS Version 4.0 through Version 5.0. Includes an appendix of DCL commands, RTL routines, and utilities eliminated from VMS Version 4.0.

OpenVMS Programming Environment Manual

Provides a general description of Compaq products and tools that define the programming environment. Introduces facilities and tools such as the compilers, the linker, the debugger, the System Dump Analyzer, system services, and routine libraries.

OpenVMS Programming Interfaces: Calling a System Routine

Describes the OpenVMS programming interface and defines the standard conventions to call an OpenVMS system routine from a user procedure. The Alpha and VAX data type implementations for various high-level languages are also presented in this manual.

OpenVMS RTL DECTalk (DTK\$) Manual

Documents the DECTalk support routines contained in the DTK\$ facility of the OpenVMS Run-Time Library.

OpenVMS RTL Parallel Processing (PPL\$) Manual

Documents the parallel-processing routines contained in the PPL\$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS Software Overview

Provides an overview of the OpenVMS operating system and some of its available products.

OpenVMS SUMSLP Utility Manual

Describes how to use the SUMSLP batch-oriented editor to update source files.

OpenVMS System Messages and Recovery Procedures Reference Manual

Contains an alphabetical listing of the errors, warnings, and informational messages issued by the operating system. Also provides the meaning of each message and a statement of the action to be taken in response to each message. This manual is in two volumes.

OpenVMS Terminal Fallback Utility Manual

Describes how to use the Terminal Fallback utility to manage the libraries, character conversion tables, and terminal parameters that are available within this utility.

OpenVMS VAX Card Reader, Line Printer, and LPA11-K I/O User's Reference Manual

Describes the card reader, laboratory peripheral accelerator, and line printer drivers on OpenVMS VAX.

OpenVMS VAX Device Support Manual

Describes how to write an OpenVMS VAX driver for a device not supplied by Compaq.

OpenVMS VAX Device Support Reference Manual

Provides the reference material for the *OpenVMS VAX Device Support Manual* by describing the data structures, macros, and routines used in device-driver programming.

OpenVMS VAX Patch Utility Manual

Describes how to use the Patch utility to examine and modify executable and shareable OpenVMS VAX images.

OpenVMS Wide Area Network I/O User's Reference Manual

Describes the DMC11/DMR11, DMP11 and DMF32, DR11-W and DRV11-WA, DR32, and asynchronous DDCMP interface drivers on OpenVMS VAX.

PDP-11 TECO User's Guide

Describes the operating procedures for the PDP-11 TECO (Text Editor and Corrector) program.

POLYCENTER Software Installation Utility User's Guide

Provides information on the POLYCENTER Software Installation utility, a new component that lets you install and manage software products that are compatible with the utility.

TCP/IP Networking on OpenVMS Systems

Provides an introductory overview of TCP/IP networking and describes OpenVMS DCL support for TCP/IP capabilities.

A

`$ABORT_TRANS` system service, 5-5, 5-16
Access date file attribute, 4-8, 5-11
`$ACK_EVENT` system service, 5-5, 5-16
ACME
 `$ACM` system service, 4-2, 5-16
 DCL commands for, 4-2
 persona items for, 3-4
`$ADD_BRANCH` system service, 5-5, 5-16
AGP support on AlphaServers, 3-2
Alias file link, definition, 4-10
ANALYZE/PROCESS command, 3-2
`$ASSIGN` system service, 5-17
AST, 1-2, 5-12
Availability Manager, 4-25

C

Clusters, see OpenVMS Cluster systems
Case sensitivity
 ACP/QIO and directory entries, 4-13
 and Compaq C RTL, 4-14, 5-2
 system services for controlling, 4-13
 using RENAME to change case, 4-14
 using SET PROCESS/CASE_LOOKUP for matching, 4-12
CD-R and CD-RW (CD-Recordable) drives, 4-1
CDRECORD, sources, 6-5
CD-ROM (compact disc read-only memory), creating your own, 4-1
CDSA, 5-6
CDU\$TABLE_SPACE logical name, 3-3, 3-4
Checksumming, on individual LAN circuits, 4-25
Chinese, support for in World-Wide PostScript Printing Subsystem (WWPPS), 3-5
Circuits, displaying information about, 4-27, 4-31
Common Data Security Architecture
 See CDSA
Compaq C RTL
 EV6 optimized C RTL, 1-3
 feature switches, 4-14, 5-2
 new functions, 5-5
 ODS-5 support, 4-14, 5-2
 POSIX style identifiers, 5-4

Compaq Secure Web Browser for OpenVMS Alpha, 8-5
Compaq SSL, using with LDAP, 5-7
COPY/FTP command, 3-2
CPU
 interrupt, 4-7
 primary and preferred, 4-7
 specifying for I/O post processing, 1-2, 4-6
`$CPU_TRANSITION` system service, 5-17
CREATE/DIRECTORY command, 3-2
`$CREATE_UID` system service, 5-5, 5-16
`$CREMBX` system service, 5-10, 5-17
C RTL
 See Compaq C RTL
Cryptography
 and Compaq SSL, 5-6
 and Kerberos, 4-16

D

Data modified file attribute, 4-8, 5-11
DCL, case sensitivity, 4-14
DCL commands
 ANALYZE/PROCESS, 3-2
 COPY/FTP, 3-2
 CREATE/DIRECTORY, 3-2
 DELETE, 3-2
 DIRECTORY, 4-9
 DUMP/HEADER, 4-9
 INITIALIZE, 3-2, 4-9, 4-11
 INITIALIZE/QUEUE, 3-2, 4-3
 MOUNT, 3-2
 PURGE, 3-2
 READ, 3-2
 RECALL, 3-2
 RENAME, 4-14
 SET/QUEUE, 4-3
 SET AUDIT, 3-2
 SET COMMAND, 3-3
 SET CPU, 3-3
 SET DEVICE, 4-8, 4-17, 4-24
 SET ENTRY, 3-3
 SET FILE, 4-11
 SET MAGTAPE, 3-3
 SET PROCESS, 4-12
 SET QUEUE, 3-3
 SET SERVER, 3-3

DCL commands (cont'd)

- SET VOLUME, 3-3, 4-9, 4-11
- SHOW/CPU, 3-3
- SHOW DEVICES, 3-3, 4-7, 4-17
- SHOW ENTRY, 3-3
- SHOW NETWORK, 3-3
- SHOW QUEUE, 3-3
- SHOW SERVER, 3-3
- SPAWN, 3-3, 4-28
- START/CPU, 3-3
- START/QUEUE, 3-3, 4-3
- STOP, 3-3
- STOP/CPU, 3-3
- SUBMIT, 3-3

DCL CTLFLAGS system parameter, 3-3, 4-28, 4-29

DECC\$SHR, 1-3

\$DECLARE_RM system service, 5-5, 5-16

DECram for OpenVMS, specifying CPU for I/O

- post processing, 4-6

DELETE command, 3-2

\$DELPRC system service, 5-17

DELPRC_EXIT system parameter, 4-29

Device drivers

- debugging, 9-5
- DECram, 4-6
- DUdriver, 4-22
- Fibre Channel, 4-22
- in a cluster, 4-21
- Mailbox, 4-6
- PEdriver, 1-2, 4-22, 4-25
- PKAdriver, 4-32
- PMdriver, 4-22
- PNdriver, 4-22
- supplied with OpenVMS, 9-5
- writing, 4-6, 9-4, 9-7

DIRECTORY command, 4-9

DUMP/HEADER command, 4-9

E

Encryption

- and Compaq SSL, 5-7
- and Kerberos, 4-16
- and Stunnel, 6-5

\$END_BRANCH system service, 5-5, 5-16

\$END_TRANS system service, 5-5, 5-16

\$ERAPAT system service, 5-17

EV6 optimized C RTL, 1-3

EXIT command, 5-13

Extended File Specifications, 4-20, 5-2

F

F\$CVTIME lexical function, 3-3

F\$GETDVI lexical function, 3-3

F\$GETJPI lexical function, 3-4, 4-14

F\$GETQUI lexical function, 3-4

F\$GETSYI lexical function, 3-4

Failover

- disabling and enabling, 4-30
- in an OpenVMS Cluster, 4-23
- specifying automatic interval for, 4-30

Fast Path, and KZPEA adapter, 4-32

Fibre Channel configuration

- and I/O interrupts, 4-23
- driver optimizations, 4-22
- failover to an MSCP-served path, 4-23
- multipath tape support, 4-24
- path balancing, 4-24
- performance improvements, 4-20

File access information extended, 4-8, 5-11

File attributes, list of, 4-8

File links, types of, 4-10

File names

- and TCP/IP, 6-3
- case sensitivity, 4-12, 4-13
- extended character support, 4-20
- UNIX style and Compaq C RTL, 5-2

Files, zipping and restoring, 6-5

\$FORGET_RM system service, 5-5, 5-16

FRED block, 1-2, 5-8

G

Galaxy

- configuring with GCM, 4-16
- DECram support, 4-6
- new F\$GETSYI items for, 3-4
- write-shared images, 5-10

\$GETDTI system service, 5-5, 5-16

\$GETDVI system service, 5-17

\$GETJPI system service, 1-2, 4-13, 4-14, 5-17

\$GETQUI system service, 5-17, 5-18

\$GETSYI system service, 5-18

\$GETTIM system service, 5-17

\$GETUAI system service, 5-18

\$GET_DEFAULT_TRANS system service, 5-5, 5-16

GNV Early Adopter's Kit, 6-5

Graphics support, 3-2

H

Hard file link, definition, 4-10

Hardware interrupts, redirecting, 4-7

HELP/MESSAGE facility, 8-6

HELP command, 8-6

I

`$ICC_CONNECT` system service, 5–18
`INITIALIZE/QUEUE` command, 3–2, 4–3
`INITIALIZE` command, 3–2, 4–9, 4–11
Interrupt CPU, specifying, 4–7
`IOC$ALTREQCOM` routine, 4–6
`IOC$REQCOM` routine, 4–6
`IOLOCK8` spinlock, 1–2, 4–23
`IOPOST_LOCAL` bit, 4–6
ISO Latin-1 character set, 4–20

J

`$JOIN_RM` system service, 5–5, 5–16

K

KZPEA adapter, 4–32

L

LDAP, and Compaq SSL, 5–7

Lexical functions

- `F$CVTIME`, 3–3
- `F$GETDVI`, 3–3
- `F$GETJPI`, 3–4, 4–14
- `F$GETQUI`, 3–4
- `F$GETSYI`, 3–4

`LOAD DSF` command, 5–13

Logical names

- `CDU$TABLE_SPACE`, 3–3, 3–4
- `SYS$POSIX_ROOT`, 5–3

M

Mailbox, performance improvements, 1–2, 5–10

Mailbox driver, specifying CPU for I/O post processing, 4–6

`MAILBOX` spinlock, 1–2

Modify date file attribute, 4–8, 5–11

Modular Data Router (MDR), 4–24

`MOUNT` command, 3–2

Mozilla, 2–3, 8–5

`MPDEV_AFB_INTVL` system parameter, 4–30

`MPDEV_REMOTE` system parameter, 4–23

N

NAML block, new `NAML$V_CASE_LOOKUP` field, 4–13

Network block count, 3–2

NUMA environment, system service updates for batch job support, 5–18

O

ODS-5 system disk, 3–3, 4–8, 4–20

Open Source Security for OpenVMS, 5–6

OpenVMS Cluster systems

- and TCP/IP enhancements, 6–3
- Fibre Channel driver optimizations, 4–22
- managing circuits and ports, 4–27
- managing interconnects, 4–30
- MSCP served disks limit, 4–23
- multipath balancing, 4–24

OpenVMS Migration Software, 6–5

P

PATHWORKS for OpenVMS, replaced by

- Advanced Server for OpenVMS on Alpha, 6–1

`PEdriver`, checksumming, 1–2, 4–25

Persona codes, 3–4

`PKAdriver`, 4–32

Ports, displaying information about, 4–27, 4–31

POSIX-style root, 5–3

Primary CPU, and preferred CPU, 4–7

Primary file link, definition, 4–10

`$PROCESS_AFFINITY` system service, 5–18

`$PROCESS_CAPABILITIES` system service, 5–18

`PURGE` command, 3–2

`$PURGWS` system service, 5–18

R

Resource Affinity Domain, See RAD

RAD

- and DECram, 4–6

- and distributed interrupts, 4–8

- displaying information for a queue, 3–3

- in a NUMA environment, 4–3

- new flag for `$GETQUI`, 5–19

- new item codes for, 5–19

- `/RAD` qualifier, 3–2, 3–3, 4–3, 4–4, 4–29

`READ` command, 3–2

`RECALL` command, 3–2

`$REGISTRY` system service, 5–18

`RENAME` command, 4–14

`RESERVED_MEMORY` commands, `ADD`,
`EXTEND`, `LIST`, and `MODIFY`, 4–29

RMS

- global buffer locking, 1–2

- new field in NAML block, 4–13

- write-behind option, 5–12

- XABDAT structure extended, 5–11

`RMS_SEQFILE_WBH` system parameter, 4–30

S

- Secure Sockets Layer, See Compaq SSL
- SSL, See Compaq SSL
- System Communications Services, see SCS
- SOS1 space, 1–2
- SCACP utility, 4–25
- SCACP utility commands
 - SET CIRCUIT, 4–31
 - SET PORT, 4–31
 - SET VC, 4–32
 - SHOW CIRCUIT, 4–31
 - SHOW PORT, 4–31
- SCHEM spinlock, 1–2
- SCS
 - assigning connections, 4–25
 - circuit load class value, 4–26
 - performance improvements, 4–22
- SCSI configuration, failover to an MSCP-served path, 4–23
- SDA commands
 - SHOW PROCESS, 5–13
 - SHOW RESOURCES, 5–13
 - SHOW TQE, 5–12
 - VALIDATE TQE, 5–12
- SDA Extended File Cache (XFC) commands
 - EXIT, 5–13
 - LOAD DSF, 5–13
 - SHOW CONTEXT, 5–13
 - SHOW EXTENT, 5–13
 - SHOW FILE, 5–13
 - SHOW MEMORY, 5–14
 - SHOW SUMMARY, 5–14
 - SHOW TABLES, 5–14
 - SHOW TRACE, 5–15
 - SHOW VOLUME, 5–15
- Security features
 - ACME, 4–2
 - CDSA, 5–6
 - Compaq SSL, 5–6
 - Kerberos, 4–16
- SET/QUEUE command, 3–3, 4–3
- SET AUDIT command, 3–2
- SET CIRCUIT command, 4–31
- SET COMMAND command, 3–3
- SET CPU command, 3–3
- SET DEVICE command, 4–8, 4–17, 4–24
- \$SETDTI system service, 5–5, 5–16
- SET ENTRY command, 3–3
- SET FILE command, 4–11
- SET MAGTAPE command, 3–3
- SET PORT command, 4–31
- SET PROCESS command, 4–12
- SET SERVER command, 3–3
- SET VC command, 4–32
- SET VOLUME command, 3–3, 4–9, 4–11
- \$SET_DEFAULT_TRANS system service, 5–5, 5–16
- \$SET_PROCESS_PROPERTIES system service, 4–13
- /SHADOW_MEMBER qualifier, 5–1
- SHOW/CPU command, 3–3
- SHOW CIRCUIT command, 4–31
- SHOW CONTEXT commands, 5–13
- SHOW DEVICES command, 3–3, 4–7, 4–17
- SHOW ENTRY command, 3–3
- SHOW EXTENT command, 5–13
- SHOW FILE command, 5–13
- SHOW MEMORY command, 5–14
- SHOW NETWORK command, 3–3
- SHOW PORT command, 4–31
- SHOW PROCESS command, 5–13
- SHOW QUEUE command, 3–3
- SHOW RESOURCES command, 5–13
- SHOW SERVER command, 3–3
- SHOW SUMMARY command, 5–14
- SHOW TABLES command, 5–14
- SHOW TQE command, 5–12
- SHOW TRACE command, 5–15
- SHOW VOLUME command, 5–15
- \$SNDJBC system service, 5–18, 5–19
- SPAWN command, 3–3, 4–28
- Spawning a subprocess, 4–28
- Spinlock changes, 1–2
- SPL SHOW COLLECT command, 5–16
- SPL SHOW TRACE command, 5–16
- START/CPU command, 3–3
- START/QUEUE command, 3–3, 4–3
- \$START_BRANCH system service, 5–6, 5–16
- \$START_TRANS system service, 5–6, 5–16
- STOP/CPU command, 3–3
- STOP command, 3–3
- Stunnel, 6–5
- SUBMIT command, 3–3
- Subprocesses, spawning and naming, 4–28
- \$SUSPND system service, 5–18
- SYS\$POSIX_ROOT logical name, 5–3
- SYSMAN utility commands
 - RESERVED_MEMORY ADD, 4–29
 - RESERVED_MEMORY EXTEND, 4–29
 - RESERVED_MEMORY LIST, 4–29
 - RESERVED_MEMORY MODIFY, 4–29
- System libraries, decompressing, 4–17
- System parameters
 - DCL_CTLFLAGS, 3–3, 4–28, 4–29
 - DELPRC_EXIT, 4–29
 - MPDEV_AFB_INTVL, 4–30
 - MPDEV_REMOTE, 4–23
 - RMS_SEQFILE_WBH, 4–30
- System services
 - \$ABORT_TRANS, 5–5, 5–16
 - \$ACK_EVENT, 5–5, 5–16

System services (cont'd)

- \$ACM, 4-2, 5-16
- \$ADD_BRANCH, 5-5, 5-16
- and case sensitivity, 4-13
- \$ASSIGN, 5-17
- \$CPU_TRANSITION, 5-17
- \$CREATE_UID, 5-5, 5-16
- \$CREMBX, 5-10, 5-17
- \$DECLARE_RM, 5-5, 5-16
- \$DELPRC, 5-17
- \$END_BRANCH, 5-5, 5-16
- \$END_TRANS, 5-5, 5-16
- \$ERAPAT, 5-17
- for DECdtm, 5-5
- \$FORGET_RM, 5-5, 5-16
- \$GETDTI, 5-5, 5-16
- \$GETDVI, 5-17
- \$GETJPI, 1-2, 4-13, 5-17
- \$GETQUI, 5-17, 5-18
- \$GETSYI, 5-18
- \$GETTIM, 5-17
- \$GETUAI, 5-18
- \$GET_DEFAULT_TRANS, 5-5, 5-16
- \$ICC_CONNECT, 5-18
- \$JOIN_RM, 5-5, 5-16
- \$PROCESS_AFFINITY, 5-18
- \$PROCESS_CAPABILITIES, 5-18
- \$PURGWS, 5-18
- \$REGISTRY, 5-18
- \$SETDTI, 5-5, 5-16
- \$SET_DEFAULT_TRANS, 5-5, 5-16
- \$SET_PROCESS_PROPERTIES, 4-13
- \$SNDJBC, 5-18, 5-19
- \$START_BRANCH, 5-6, 5-16
- \$START_TRANS, 5-6, 5-16
- \$SUSPND, 5-18
- \$TRANS_EVENT, 5-6, 5-16

T

Tarfiles, program for reading and writing, 6-5

Timer Queue

- change in format, 1-2
- new TQE type, 5-12

\$TRANS_EVENT system service, 5-6, 5-16

U

UCB\$L_STS field, 4-6

Unit control block (UCB) initialization routine, 4-6

UNIX

- file access information, 5-11
- file name support, 4-8
- file name support and Compaq C RTL, 5-2
- path name support, 6-3
- portability, 4-8
- porting tools, 6-5

UNIX (cont'd)

- style root, 5-3

V

VALIDATE TQE command, 5-12

VMSTAR, 6-5

Volume Shadowing

- specifying CPU for I/O post processing, 1-2, 4-6

- specifying dump file information, 5-1

W

Web browser, 2-3, 8-5

Wildcards, use of, 3-4

X

XA standard interface, implemented in DECdtm, 4-32

Z

Zip utility, 6-5

